Analysis and implementation of discrete-time sliding mode control

Olivier Huber

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : Automatique et Productique

Arrêté ministériel : 7 août 2006

Présentée par

Olivier HUBER

Thèse dirigée par Bernard BROGLIATO
et co-encadrée par Vincent ACARY

préparée au sein de l'INRIA
et de l'école doctorale EEATS

Analyse et implémentation du contrôle par modes glissants en temps discret

Thèse soutenue publiquement le 5 mai 2015,
devant le jury composé de :

Mr Franck PLESTAN Professeur des Universités – École Centrale de Nantes,	Président
Mr Jean-Pierre BARBOT Professeur des Universités – IPGP,	Rapporteur
Mr Jamal DAAFOUZ Professeur des Universités – Université de Lorraine,	Rapporteur
Mr Gildas BESANÇON Professeur des Universités – Grenoble INP,	Examinateur
Mr Wilfrid PERRUQUETTI Professeur des Universités – École Centrale de Lille,	Examinateur
Bernard BROGLIATO Directeur de Recherche – INRIA Rhône-Alpes,	Directeur de thèse
Vincent ACARY Chargé de Recherche – INRIA Chile,	Encadrant de thèse

Contents

Acronyms

AVI	Affine Variational Inequality.
CP	Complementary Problem.
DI	Differential Inclusion.
ECB-SMC	Equivalent-Control Based Sliding Mode Control.
GE	Generalized Equation.
HOSM	Higher-Order Sliding Mode.
HOSMC	Higher-Order Sliding Mode Controller.
LCP	Linear Complementary Problem.
lsc	lower semi-continuous.
LTI	Linear Time Invariant.
ODE	Ordinary Differential Equation.
SMC	Sliding Mode Control.
VI	Variational Inequality.
ZOH	Zero-Order Hold.

List of Symbols

$F\colon X \rightrightarrows Y$	Multivalued (or set-valued) mapping from X to the set of all subsets of Y (2^Y).
I_n	Identity matrix of size $n \times n$.
$\overline{\mathbb{R}}$	Extended real line: $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$.
$\operatorname{dom} f$	Effective domain: $\{x \in \mathbb{R}^p \mid f(x) < +\infty\}$.
$\Pi_K(\cdot)$	Projector mapping onto the set K or the subspace spanned by the matrix K.
$\operatorname{Sgn}(\cdot)$	Multivalued signum function: $\begin{array}{l}\operatorname{Sgn}(x) = \{1\} \text{ if } x > 0, \operatorname{Sgn}(x) = \{-1\} \text{ if } x < 0,\\ \operatorname{Sgn}(x) \in [-1, 1] \text{ if } x = 0\end{array}$.
$\delta_K(\cdot)$	Indicator function of the set K: if $x \in K$, $\delta_K(x) = 0$, else $\delta_K(x) = +\infty$.
$\mathcal{A}(\cdot)$	Maximal monotone operator.
$\mathfrak{h}_K(\cdot)$	Support function of the set K: $\mathfrak{h}_K(x) = \sup_{y \in K} \langle y, x \rangle$.
$\partial f(\cdot)$	Subdifferential of the function f: $\partial f(x) := \{v \mid f(y) - f(x) \geq \langle v, y - x \rangle\}$.
$\mathbb{1}$	Vector of ones.
$\mathcal{B}_p$	Unit ball centered at the origin for the p-norm.
$N_K(x)$	Normal cone to K at x: $\{y \in \mathbb{R}^p \mid \langle y, z - x \rangle \leq 0, \quad \forall z \in K\}$.
O	Big-O notation: $x = O(h^p)$ as $h \to 0$ if and only if $\lim_{h \to 0} \frac{x}{h^p} = c \in \mathbb{R}$.
$\operatorname{diag}\{d_i\}$	Diagonal matrix with diagonal elements d_i.
$\overline{\operatorname{co}}\, K$	Convex Hull of the set K.

Introduction

Quick overview of the Sliding Mode Control field

Amongst the earliest documented (and accessible) studies on sliding motion, we have some works from the German school by Flügge-Lotz [39] and the two papers of André and Seibert [11, 12]. But the main developments definitively come from the Russian/Soviet school, with the first research activities also dating back to the 50's [7, 105]. Those research efforts articulate around the discontinuous aspect of the control.

Contributions from various fields have to be acknowledged: for instance the evolution of dynamical systems with a discontinuous control can be described as an Ordinary Differential Equation (ODE) with discontinuous right-hand side. This topic is shared with the optimal control theory and thus the theory of Sliding Mode Control (SMC) greatly benefited from the advances in the latter field. The works masterly summarized in Filippov's book [38] are a good example of such cross-fertilization.

In the 70's, the theory of SMC has matured and began to get known worldwide thanks to Itkis's book [63] in 1976 and the review paper by Utkin [106] in 1977. The latter also published a book on sliding modes in 1981 in Russian, later translated in English [107]. All the developments up to then deal with what we refer to as "conventional" or "classical" SMC.

The next major milestone in the sliding mode field is the introduction of Higher-Order Sliding Mode (HOSM) by Levant, first in [34] but for the most part started with [73]. This sparked the development of a large wealth of literature at the end of the 90's and in the years 2000's. The HOSM concept was applied to controllers, observers and differentiators. A recent account of the development in both "conventional" or "classical" SMC and in the HOSM field is given in [99].

However, one topic is usually left out, as mentioned in [99, p. 99]: the discrete-time case. By this term, we refer to the setup where the control input can only change at isolated

time instants t_k and the dynamical system we want to control is a continuous-time process. In our context, this means that the control input is constrained to be a step function. Let us motivate why it is interesting to study this case. Firstly this case appears when the controller is digitally implemented, for instance with the help of a microcontroller. This kind of setup is nowadays ubiquitous in benchmarks and industrial applications. Secondly We also face this situation in numerical simulation. We shall state that we were inspired by the research effort in nonsmooth mechanics to properly simulate some systems like those with dry friction and/or unilateral constraints [64, 17].

Topics tackled in this work

This thesis started in the wake of the work of my two advisors, which can be found in [4] and [5]. The main research effort is to work on the undesired phenomenon that systems with SMC exhibit: the chattering. We characterize it as the fact that the state of the system chatters in a neighborhood of the sliding manifold instead of converging on it and that the control input is of the bang-bang type. In contrast to previous approaches, we single out the chattering that is already seen in simulation, even with no disturbance and with perfect knowledge of the dynamics. We refer to this one as the *numerical* chattering and one of its distinct feature is the constant chattering, or high-frequency bang-bang behavior, of the control input. This naturally induces a chattering of the sliding variable. We claim that this type of chattering is usually predominant and that it is due to a bad (purely explicit) discretization of the Sgn multifunction. We further discuss this topic in Section 1.3. Let us now summarize the contribution of this thesis in the three main domains of control theory[1]:

ANALYSIS We restrict ourselves to two controllers in discrete-time setting. Firstly for the Equivalent-Control Based Sliding Mode Control (ECB-SMC), proofs of finite-time Lyapunov stability for the sliding variable dynamics are given. Those apply to a fairly generic class of systems, which was not the case of previous approaches. The robustness of the controller is also investigated, as well as Sliding Mode Observers and an extension to enhance the perturbation attenuation. Regarding Higher-Order Sliding Mode Controller (HOSMC), the implicit discretization of the *twisting* controller is studied. This prompted us to modify the discrete-time control input. A Lyapunov function can then be used to prove finite-time stability. Those topics form the bulk of Chapter 2.

[1] in the author's view

SIMULATION We present a Control toolbox implemented in SICONOS, a platform developed at INRIA, and various algorithms to solve the optimization problems arising from the control input computation. This part is briefly mentioned at the beginning of Chapter 3, with some highlights on how to architect a simulation software to have reliable and faithful simulations. Then we present some simulation results illustrating various analytical results along with a numerical comparison of discretization schemes in the rest of this chapter.

EXPERIMENTS Chapter 4 is devoted to the presentation and analysis of the experimental data collected on two experimental benchmarks: an electropneumatic actuator in Nantes and an inverted pendulum on a cart in Lille. On the first setup, we implemented a classical first-order sliding mode controller, as well as the twisting algorithm. On the second one, only a first-order SMC was implemented. The results of those experiments sustain the superiority of the implicit discretization introduced as shown in Chapter 2.

Before moving to the topics mentioned above, Chapter 1 is dedicated to the introduction of the topic and also the tools used in Chapter 2.

What is not discussed in this work

Given how large the Sliding Mode Control field is, we have to focus on a subset of possible research directions. In particular, we would like to stress that we will not discuss the following topics:

- The design of the sliding surface: we mostly deal with the case where the latter is a subspace of the state ($\sigma = Cx$), but we do not detail how the matrix C is chosen. We only require that the dynamics on the sliding surface ($\sigma = 0$) is asymptotically stable.

- Some classes of controllers like Integral or Terminal SMC and most HOSM controllers: we only study the classical SMC, with an equivalent control input. In the HOSM controller class, we study only the twisting algorithm. The controllers studied in this work fall into the square "fully discontinuous" category. The latter is informally defined as the fact that the control input is only a discontinuous function of the state, except for the equivalent part of the control, and the square part means that there are as many discontinuous terms as the size of the sliding variable. Fully

discontinuous controllers can be effectively modeled using the chosen frameworks (Convex Analysis and Variational Inequality).

– We do not consider relay systems where the control input can only take value in a finite discrete set. Filippov's framework is also invoked in this case to regularize the solution and understand what would be the behavior on the sliding manifold with relaxed control. We consider that the control input is multivalued ($u\colon \mathbb{R}^p \rightrightarrows U$) and takes value in a compact convex set U. Controllers with a discrete value control set are an ongoing research topic, see for instance [54, 78].

Part of the results presented here lead to the following publications:

- O. Huber, V. Acary, B. Brogliato, and F. Plestan, *Discrete-time twisting controller without numerical chattering: analysis and experimental results with an implicit method*, in Decision and Control, 2014 53th IEEE Conference on, IEEE, pp. 4373–4378.

- O. Huber, V. Acary, and B. Brogliato, *Enhanced matching perturbation attenuation with discrete-time implementations of sliding-mode controllers*, in Control Conference (ECC), 2014 European, IEEE, pp. 2606–2611.

- O. Huber, V. Acary, B. Brogliato, and F. Plestan, *Comparison between explicit and implicit discrete-time implementations of sliding-mode controllers*, in Decision and Control, 2013 52th IEEE Conference on, IEEE, pp. 2870-2875.

Chapter 1

Background Material

We now present some background material on topics used mostly in Chapter 2. Firstly, we shall quickly recall the basic theory of SMC, before touching on HOSMC. Then we skim through the mathematical tools for the analysis of ODE with discontinuous right-hand side and Differential Inclusion (DI). Finally we move to the time-discretization of SMC and also some ODE, and we present a short literature review of this topic.

1.1 Sliding Mode Control in Continuous Time

In this brief exposition of basic SMC theory, let us consider the linear case, that is with a Linear Time Invariant (LTI) system and a sliding variable defined as a linear subspace of the state. Hence in the sequel, we consider well-posed linear systems (in the sense of Filippov [38]) of the form

$$\begin{cases} \dot{x}(t) = \mathcal{A}x(t) + Bu(t) + B\xi(t) \\ u_{\text{cont}}(t) = u_{\text{cont}}^{eq}(t) + u_{\text{cont}}^{s}(t) \\ \sigma(t) \coloneqq Cx(t) \\ -u_{\text{cont}}^{s}(t) \in \alpha \operatorname{Sgn}\left(\sigma(x(t))\right), \end{cases} \tag{1.1.1}$$

with $x(t) \in \mathbb{R}^n$, $u_{\text{cont}}(t) \in \mathbb{R}^p$, $\sigma(t) \in \mathbb{R}^p$, $C \in \mathbb{R}^{p \times n}$, and $\alpha > 0$. The matched disturbance is denoted as ξ and if the system is *nominal* then $\xi \equiv 0$. Let us assume that C is chosen such that the triplet $(\mathcal{A}, B, C)$ has a strict vector relative degree $(1, 1, \ldots, 1)$. This implies that the "decoupling matrix" CB is full rank.

1.1.1 The Equivalent Control Based Sliding Mode Controller

With the ECB-SMC variant, the control input is split in two parts: the smooth equivalent part u^{eq}_{cont} which makes the sliding surface invariant (and in linear case, any level set of σ). The discontinuous part u^{s}_{cont} which brings the system to the sliding manifold and ensure the rejection of matched perturbations. Let us now derive the expression for the two control inputs. For u^{eq}_{cont}, we start from the dynamics of the sliding variable in the nominal system (1.1.1):

$$\dot{\sigma}(t) = CAx(t) + CBu^{eq}(t) + CBu^{s}(t).$$

The control law u^{eq}_{cont} is designed such that the system stays on the sliding surface once it has been reached (in other word u^{eq}_{cont} makes the sliding surface invariant with $u^{s}_{\text{cont}} \equiv 0$):

$$\dot{\sigma}(t) = 0 \text{ and } u^{s}_{\text{cont}}(t) = 0 \quad \Rightarrow \quad u^{eq}_{\text{cont}}(t) = -(CB)^{-1}CAx(t). \tag{1.1.2}$$

With this equivalent controller, the sliding variable dynamics reduces to

$$\begin{cases} \dot{\sigma}(t) = CBu^{s}(t) \\ -u^{s}_{\text{cont}}(t) \in \alpha\,\text{Sgn}(\sigma(t)). \end{cases} \tag{1.1.3}$$

The dynamics in (1.1.1) can be rewritten as

$$\dot{x}(t) = (I - B(CB)^{-1}C)Ax(t) + Bu^{s}(t),$$

or equivalently

$$\dot{x}(t) = \Pi_{\ker C}Ax(t) + Bu^{s}(t), \tag{1.1.4}$$

with $\Pi_{\ker C} \coloneqq I - B(CB)^{-1}C$. Two interesting properties of $\Pi_{\ker C}$ are $C\Pi_{\ker C} = 0$ and $\Pi_{\ker C}$ is a projector onker C [33]. The regularity of x enables us to take the equivalent integral representation of system (1.1.4)

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^{t} \Phi(t, \tau)Bu^{s}(\tau)\mathrm{d}\tau,$$

with $\Phi(t, t_0) = e^{\Pi_{\ker C}A(t-t_0)}$ the state transition matrix for the system (1.1.4). Some of the properties of $\Phi(\cdot, \cdot)$ are given in the following lemma.

Lemma 1.1.1. *One has* $\dot{\Phi}(t, t_0) = \Pi_{\ker C}A\Phi(t, t_0)$, $\Phi(t_0, t_0) = I$, *and* $C\Phi(t, t_0) = C$ *for all* $t \geq t_0$.

Proof. One has $C\dot{\Phi}(t, t_0) = 0$ so $C\Phi(t, t_0) = C\Phi(t_0, t_0) = C$ for all $t \geq t_0$. □

The sliding surface C has to fulfill only one requirement: the "zero dynamics" on the sliding manifold have to be stable. That is the dynamics of x with $\sigma = 0$. This can be for instance achieve using a pole placement technique, see for instance [6, 107], or using other principles, like in [92] where the effects of unmatched perturbations is reduced. We now quickly recall the main points that made Sliding Mode Control popular.

FINITE-TIME LYAPUNOV STABILITY OF THE AUXILIARY SYSTEM

Let us study the stability of the system (1.1.3) with the Lyapunov functions proposed in [107]. First we have the quadratic one $V(\sigma) = \sigma^T (CB)^{-1}\sigma$, which is decreasing if CB is symmetric positive-definite. The symmetry condition can be relaxed by considering the function $V(\sigma) \coloneqq \|\sigma\|_1 = -\mathcal{U}^s_{\text{cont}}\sigma$. Both can be used to show that if the system is stable, σ goes to 0 in finite-time. In Section 2.2.2, we show that the same function, only slightly modified, can be used to prove the finite-time Lyapunov stability of the discrete-time auxiliary system under similar conditions.

REJECTION OF MATCHED PERTURBATIONS

Besides finite-time stability, the rejection of matched perturbation makes SMC very attractive. In the celebrated paper [31], the concept of matched perturbation is introduced. Given a large enough control action, the perturbation has no action on the system if the latter is in the sliding phase. The matched condition basically states that the perturbation acts in the same subspace as the control input. In the linear case, with a perturbation of the form $F\Gamma$, $F \in \mathbb{R}^{n\times p}$ and $\Gamma\colon \mathbb{R} \to \mathbb{R}^p$, this condition amounts to

$$\operatorname{rank}[B, F] = \operatorname{rank} B.$$

Given that this relation holds, we know that each column of F is a linear combination of columns of B. Hence there exists some matrix $Q \in \mathbb{R}^{p\times p}$ such that $B = FQ$. This justifies the expression $B\xi$ to denote the perturbation. In discrete-time we shall see that the perturbation is not perfectly rejected, but rather its action is attenuated by the controller. This is the topic of Section 2.2.2 and in Section 2.5 we propose an extension to enhance the perturbation attenuation.

1.1.2 Higher Order Sliding Mode

We shall here give a brief account of the concept of HOSM introduced by Levant in [73] and later developed in [75] and many references. We introduce the twisting algorithm, which is to the best of our knowledge the only "fully discontinuous square" HOSMC and therefore can be studied with the tools we make use of here. Let us first define what we mean by "fully discontinuous square" controller

Definition 1.1.2. A controller is said to be *fully discontinuous* if its control input is multi-valued (or set-valued) and defined by a discontinuous function (usually Sgn) of the sliding variable(s) with an optional additive continuous term, the equivalent part of the control input. It is said to be *square* if the dimension of the sliding variable and the number of discontinuous terms in the control law are equal.

We allow the presence of the equivalent part of the control input since this one is defined by the selection procedure arising from Filippov's framework and therefore can be considered as part of the Sgn function. Let us formally define the set-valued variant of this function.

Definition 1.1.3 (Multivalued signum function). Let $x \in \mathbb{R}$. The multivalued signum function $\mathrm{Sgn} : \mathbb{R} \rightrightarrows \mathbb{R}$ is defined as:

$$\mathrm{Sgn}(x) = \begin{cases} \{1\} & x > 0 \\ \{-1\} & x < 0 \\ [-1, 1] & x = 0. \end{cases}$$

If $x \in \mathbb{R}^n$, then the multivalued signum function $\mathrm{Sgn} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is defined as: for all $j = 1, \dots, n$, $(\mathrm{Sgn}(x))_j := \mathrm{Sgn}(x_j)$.

The material shown in the sequel is adapted from [99].

Historically, the first HOSMCs are of the second order type, that is there is a relative degree 2 between the sliding variable σ and the control input u:

$$\ddot{\sigma} = f_s(x, t) + g_s(x, t)u, \tag{1.1.5}$$

with x the state of the plant. The goal is first to bring both σ and $\dot{\sigma}$ to 0 in finite-time and then to maintain the trajectories of the system at the origin of the σ–$\dot{\sigma}$ plane. Once this is the case, a sliding motion occurs on the intersection of the hyperplanes $\sigma = 0$

and $\dot{\sigma} = 0$. The stability analysis for this kind of controllers typically requires that the following inequalities hold for some positive scalars $\aleph_m$, $\aleph_M$ and $\complement$:

$$0 < \aleph_m \leq g_s(x,t) \leq \aleph_M \quad \text{and} \quad |f_s(x,t)| \leq \complement \quad \text{for all } x \in \mathbb{R}^n \text{ and } t.$$

Now we shall see how the control input is defined as a function of σ and $\dot{\sigma}$ for the different controllers.

The twisting algorithm

Let us start with the twisting algorithm, in which the control input is given by the relation

$$-u \in G(\mathrm{Sgn}(\sigma) + \beta\,\mathrm{Sgn}(\dot{\sigma})) \quad \text{with } G, \beta > 0. \tag{1.1.6}$$

The auxiliary system (1.1.5) is globally finite-time stable if the parameters G and β satisfy the following conditions

$$G(1+\beta)\aleph_m + \complement > G(1-\beta)\aleph_M + \complement \quad \text{and} \quad G(1-\beta)\aleph_m > \complement. \tag{1.1.7}$$

A typical "phase-plot" of a dynamical system with the control input given by the twisting algorithm is given in Figure 1.1.

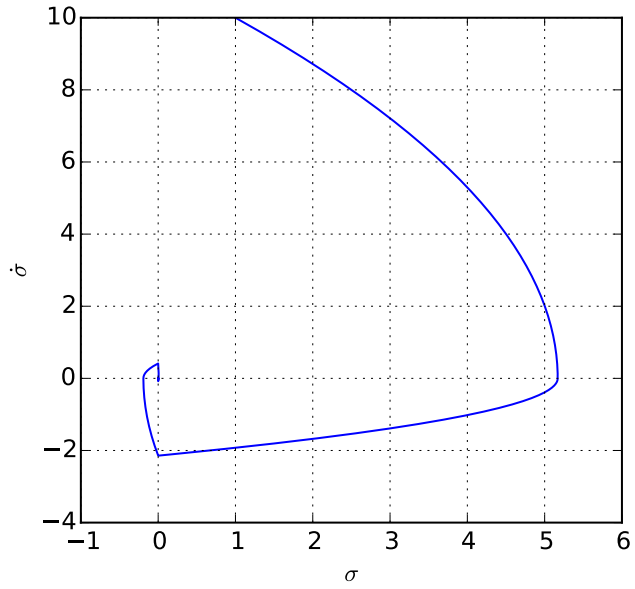


Figure 1.1: Typical evolution of the sliding variables with the twisting algorithm.

Other 2-Sliding Mode Controllers

Among the other controllers for system (1.1.5) based on the HOSM principle, we have the *suboptimal* algorithm. The control input is in this case defined as:

$$-u \in r_1 \operatorname{Sgn}(\sigma - \sigma^*/2) - r_2 \operatorname{sgn}(\sigma^*) \quad \text{with } r_1 > r_2 > 0,$$

where σ^* is the value of σ at the last time $\dot{\sigma} = 0$. Another one is the controller with *prescribed convergence law*, which has a set-valued control law given by

$$-u \in \alpha \operatorname{Sgn}(\dot{\sigma} + \xi(\sigma)) \quad \text{with } \alpha > 0.$$

Finally there is also the *quasi-continuous* controller, which defines u as:

$$u = -\alpha \frac{\dot{\sigma} + \beta|\sigma|^{1/2} \operatorname{sgn}(\sigma)}{|\dot{\sigma}| + \beta|\sigma|^{1/2}} \quad \text{with } \alpha, \beta > 0.$$

Those controllers do not fulfill the conditions of Definition 1.1.2: the first one has a term $\operatorname{sgn}(\sigma^*)$ which does not depend on a state variable. The second one has only one Sgn and the last one has no multivalued Sgn function. Now that we have defined the controllers that we study, let us expand a bit on Filippov's framework.

1.2 ODE with Discontinuous Right-Hand Side

The study of existence of a solution of an ODE with discontinuous right-hand side fails with the classical tools for ODE. If a surface of discontinuity is reached at a certain time instant and the vector fields are locally directed towards it, it is not clear how to define a solution after that instant. To overcome this difficulty, the idea is to recast the ODE as a DI, that is

$$\dot{x} = f(x, t) \quad \Rightarrow \quad \dot{x} \in F(x, t), \tag{1.2.1}$$

with $F\colon \mathbb{R}^n \times \mathbb{R} \rightrightarrows \mathbb{R}^n$ a multifunction with some desirable properties. In what follows, we require F to have compact convex images $F(x, t)$. We also require it to be upper semicontinuous (that is for all $(x_0, t_0) \in \mathbb{R}^n \times \mathbb{R}$ and open set $\mathcal{N}$ with $F(x_0, t) \in \mathcal{N}$, there exists a neighborhood $\mathcal{M}$ of (x_0, t_0) such that $F(\mathcal{M}) \subset \mathcal{N}$). With those hypothesis, existence of an absolutely continuous solution to the DI (1.2.1) is guaranteed, see for instance [13, Theorem 3, p. 98]. Absolutelt continuity is a nice property since it is equivalent to differentiability almost everywhere. Now the preeminent task is to define F from the function f. In Filippov's book [38], several proposals are listed. It is worth noting that, in general, a solution may or may not exist depending on how F is chosen. However in

this work the (simple) cases we tackle do not require any particular way of constructing F. We briefly mention three possible constructions given by Filippov at the beginning of Chapter 2 in his book.

- The convex procedure introduced by Filippov in [37]: the idea is to take the convex hull of all the vector fields in the neighborhood of the sliding surface which exists on a set with a nonzero measure at a fixed time t. Mathematically speaking, this means
$$F(x,t) = \bigcap_{\delta>0} \bigcap_{\mu(N)=0} \overline{\text{co}}\, f(x + \delta\mathcal{B} \setminus N, t).$$

- The equivalent control method, credited to Utkin: we suppose that the ODE has the form
$$\dot{x} = f(x,t,u)$$
with u a discontinuous function with each component discontinuous on a smooth S_i defined by $\varphi_i(x,t) = 0$. Let us introduce the following concept: a surface of discontinuity is given by a sliding surface S or the intersection of several sliding surfaces S_i with $i \in I$. If the point x is on a surface of discontinuity, then the equivalent control u^{eq} is computed such that the system would not leave the manifold while each component u_i of u is constrained to take value in a closed interval formed by the value of u_i on each side of surface S_i. That is the equivalent control is implicitly defined as
$$\nabla\varphi_i(x,t) f(x,t,u^{eq}) = 0 \quad \forall i \in I.$$
This is equivalent to the DI
$$\dot{x} \in f(x,t,U(t,x)) = \left\{ f(x,t,v) \mid v \in U(t,x) \right\} =: F_{eq}(x,t),$$
where the set U is defined as the Cartesian product of the intervals we just mentioned.

- The method introduced by Aizerman and Pyatnitskii [8, 9]: consider the DI
$$\dot{x} \in \overline{\text{co}}\, F_{eq}(x,t),$$
and add the constraint that on a surface of discontinuity $\mathcal{M}$, the velocity has to remain tangent to it. Then letting $K(x,t)$ be the intersection of $\overline{\text{co}}\, F_{eq}(x,t)$ and the tangent space to the $\mathcal{M}$ at x, we get
$$\dot{x} \in K(x,t).$$

Now the main question is whether we have to be careful about the method used to get F. Fortunately in our LTI context we do not have to. Considering the discussion in [38, p. 56], if f is linear in u and if all the sliding surfaces are different and at their intersection their normal vectors are linearly independent, then all methods yield the same DI. The first condition on f is already satisfied since we already restricted ourselves to the nonlinear affine in control case. For the second one, in the case of linear switching surfaces defined by a matrix C, it means that the rows of C are linearly independent. In any case, the systems we deal with in the sequel must satisfy such conditions.

1.3 The Chattering Problem

The concept of chattering is (unfortunately) paired with Sliding Mode Control but its definition is not entirely accurate. The most common explanation is behavioral in nature: the chattering is the "zigzag" motion around the sliding manifold [99, p. 8]. However this phenomenon is also tied to the high frequency switching control input that is seen in simulation and experiments. It is easy to see that if the control input has such behavior, then the state and hence the sliding variable have such a "zigzag" motion.

Several attempts to model or approach the chattering phenomenon have been made, see for instance [41, 76, 109]. However few have studied the chattering from an discretization perspective, named "discretization chattering" in [109]. We can find in the works of Galias, Yu and their collaborators [45, 46, 47, 117] a study of the relationship between the chattering and the explicit discretization. We also engage in this direction: we highlight the contribution of discretization issues to the chattering.

Definition 1.3.1. We call *numerical chattering* the fact that the control input is switching at a high frequency because of a bad discretization. Such control input induces a chattering behavior on the sliding variable.

This chattering can be seen in simulation even with no perturbation and a perfect model. The aim of this work is to suppress this numerical chattering. We would like also to add that in Control Theory, the term chattering is only used in the context of SMC. Yet the presence of noise, perturbation or unmodelled dynamics affects any control law and should also produce "chattering". We would like to underline that due to the discontinuous nature of SMC, this control technique may be the one that is the most affected by a bad discretization method. Hence chattering may be even more intimately related than it is generally thought.

We also introduce here a way to characterize quantitatively the chattering. We propose to use the *variation of a function* as a measure of chattering. Two main motivations are the fact that absolutely continuous functions are of bounded variation. Then this quantity has to remain finite and the convergence of its value as $h \to 0$ is possible. The material in the following definition is adapted from [10].

Definition 1.3.2. Let $f : \mathbb{R} \to \mathbb{R}^m$ be a right-continuous step function, discontinuous at finitely many time instants t_k and t_0, $T \in \mathbb{R}$ with $t_0 < T$. The variation of f on $[t_0, T]$ is defined as:

$$\mathrm{Var}_{t_0}^{T}(f) := \sum_{k} \left\| f(t_k) - f(t_{k-1}) \right\|,$$

with $k \in \mathbb{N}^*$ such that $t_k \in (t_0, T]$. If f is continuously differentiable with bounded derivatives then the variation of f on $[t_0, T]$ is defined as:

$$\mathrm{Var}_{t_0}^{T}(f) := \int_{t_0}^{T} \left\| \dot{f}(\tau) \right\| \mathrm{d}\tau.$$

1.4 Time Discretization

This work concentrates on discrete-time SMC, where the control input value changes only at certain time instants. In order to compute the control input value, we need to discretize the dynamics in time to get a recurrence relation describing how the system evolves in time. We shall now quickly review the different discretization methods before mentioning some work in the literature that aimed at tackle similar issues.

1.4.1 Time discretization of an ODE and discrete-time controller

The time discretization (or temporal discretization) is the action of transforming an ODE

$$\dot{x} = f(x, t)$$

into a recurrence or difference relation. For the one-step methods, the latter is of the form

$$x_{k+1} = F(x_k, x_{k+1}, t_k, t_{k+1}). \tag{1.4.1}$$

Its study belongs to the field of Numerical Analysis. The main motivation behind this transformation is the numerical simulation of dynamical systems governed by an ODE, which requires a relation similar to (1.4.1) in order to perform computations. This process

is called the numerical integration of the system. Indeed the time discretization is closely related to the problem of quadrature, that is to approximate the value of an integral. With the increased availability of computers, a large wealth of literature on this topic is now available. Major developments can be found in [50, 51, 82]. A important notion developed is the order of an integration scheme. Unfortunately there are two definitions for this concept: the first one is that a numerical integration is of order p if it can exactly integrate any polynomial of degree no greater than p. The second one is that if the error on the solution is $O(h^p)$, then the order of the scheme is p. However those two definitions are not consistent since the first one implies that the error is $O(h^{p+1})$. Here we shall use the second definition.

Two adjectives are commonly used to characterize an integration method: *explicit* and *implicit*. A method is said to be *explicit* if the right-hand side F in (1.4.1) depends only the known quantities t_k, t_{k+1} and x_k. A method is said to be (fully) *implicit* if F depends (only) on t_k, t_{k+1} and x_{k+1}. Methods can also combine explicit and implicit parts. Usually there is no closed-form formula for an implicit method: finding the next value x_{k+1} is equivalent to a root-finding problem. Most of the time a Newton-Raphson method is used, which is recognized as one of the most delicate part in the implementation of such scheme [51].

We shall highlight some peculiarities of the numerical integration of ODEs with discontinuous right-hand side, for instance those coming from contact mechanics and SMC. In general one wants to integrate an ODE with an high order method, even if the computational cost increases since the faithfulness of the simulation also increases. Then a compromise has to be found between the integration error and the computation time. However for the type of aforementioned ODEs, it is not recommended to use a method with higher order, especially for the argument of a discontinuous function like signum. It induces spurious oscillations that may jeopardize the simulation, see [3, p. 273]. For our case this is not a real restriction since the use of microcontroller to provide the control input value to the plant imposes that the control input is constant on the interval $[t_k, t_{k+1})$.

Let us illustrate the difference between the explicit and implicit discretizations of a sliding mode controller with the following academic example:

$$-\dot{x}(t) \in \alpha \operatorname{sgn}(x(t)),$$

with $\alpha > 0$. We use the differential inclusion framework as we let $\operatorname{sgn}(0)$ to take any value in $[-1, 1]$ (this is formally stated in Definition 2.2.1 as Sgn). An *explicit* discretization yields $x(t_{k+1}) \in x(t_k) - h\alpha \operatorname{sgn}(x(t_k))$ whereas the *implicit* one yields $x(t_{k+1}) \in x(t_k) - h\alpha \operatorname{sgn}(x(t_{k+1}))$. As long as $|x(t_k)| \gg h\alpha$, there is no difference between the two methods.

But if $|x(t_k)| < \alpha h$, then the behavior changes with the type of discretization. With $0 < x(t_k) < h\alpha$, in the explicit case, $x(t_{k+1}) \in x(t_k) - h\alpha \operatorname{sgn}(x(t_k)) < 0$. The state switches sign at every time instant t_k, leading to the well-known chattering phenomenon. Meanwhile in the implicit case, the control algorithm guarantees $x(t_{k+1}) = 0$ by choosing $\operatorname{sgn}(x(t_{k+1})) = x(t_k)/(\alpha h) < 1$. Hence the system reaches the origin and for all $l \geq k+1$, we have $x_l = 0$ since we can select $\operatorname{sgn}(x_l) = 0$.

1.4.2 Previous work in discrete-time Sliding Mode Control

Let us finish this short review by mentioning researches that were, in the author's eyes, going into the right direction. First of all, one of the most promising id'ea was developed in [30], for the scalar case: both the state and the control input have dimension one. The argument of the signum function is implicit even though it is not stated in those terms. In the discrete-time SMC literature, this is in our view, the closest previous work the developments in this thesis. The two authors later separately developed different approaches with an accent on the equivalent part of the control. In [108], a deadbeat-like control input is presented, with the control input being projected onto an admissible set. In [104], the sgn part of the control input is also removed, and the proposed control law includes a prediction of the control law to enhance the behavior around the sliding manifold. This inspired us the development in Section 2.5 but with a different approach: we still have a discontinuous part in the control input and the perturbation prediction is an additional term to the control input.

Other major works in the domain of discrete-time SMC are, in a chronological order, [97, 42, 48]. In the first one, the authors give a condition for the convergence of the system to the sliding manifold and also an upper and lower bound on the control input (as opposed to the continuous-time case where there is only a lower bound). They also provide an algorithm to compute the control law in the form of a linear feedback with varying gains. The paper was latter criticized in [70] and [115] for the additional requirement of an upper bound and in the second paper [115], the author claims to provide an example where the algorithm of the original paper fails to induce stable trajectories. In the second paper [42], the author proposes a discrete-time version of the equivalent control and a control law also based on a varying state feedback. The last work [48] introduces the concept of *reaching law*, which roughly speaking tries to impose the dynamics of the sliding variable and to define the control law such as to impose some performances.

The present work differs from the aforementioned approaches in the following ways:

- The accent is on the discontinuous part of the control input. We claim that it is the implicit discretization of the Sgn multifunction that is the cornerstone of a successful discrete-time SMC. The equivalent part in the control input has to be properly discretized, but this is not mandatory to get a good behavior. We also provide pointers for the analysis and design of SMC with no (explicit) equivalent control term.

- The conditions given in our stability results depend only on system or controller parameters (like the gain for the control input, the matrix multiplying the control input) instead of stating them in terms of the sliding variable evolution.

- Our results are valid for an arbitrary number of sliding variables and do not preclude coupled evolution of sliding variables.

In the last decade, the behavior of the *explicitly* discretized SMC was extensively studied, see [45, 46, 47, 111, 117, 116] and other references therein. The main research axis was to show the existence of limit cycles, defined in term of the signum of the sliding variable. This further supports our claim that the explicit discretization should not be used for the discontinuous part of the control.

Additional references on discrete-time SMC include [1, 43, 49, 61, 69, 71, 79, 84, 85, 101, 103, 111].

1.4.3 Previous work in the Control community

Looking at the control theory literature, the complementarity framework has already been used for some feedback systems. Most of this body of literature comes from the Dutch school, see [20, 21, 22, 53, 81, 91], where their mainly use the complementarity framework. In Section 2.1 we present two frameworks that can be used to analyze the discrete-time SMC and one of them is complementary. However in the present approach we decided not to use it, but rather to use Variational Inequality (VI) and we give some reasons for that choice.

1.4.4 Numerical Analysis

The discretization of differential inclusions has also some history. Major research efforts have been done in the 90's, see [29, 28, 67, 66, 72]. In those works, the discretization of DIs arising from ODEs with discontinuous right-hand side as well as general DIs is considered.

The results are mainly focused on the convergence of the discretized solution as well as the convergence order analysis. In [14], the special case of a DI with a multivalued maximal monotone term is studied and orders of an half-implicit numerical scheme are provided. Those results in the field of Applied Mathematics show that the use of an implicit scheme, at least on the multivalued term, is sound by considering the properties of the numerical solution.

However, to the best of our knowledge, those authors do not study the convergence of the implicitly discretized multivalued term to the continuous-time one. We believed to be the first ones to do so, until we stumbled upon [96] where this kind of study is done, also in the context of DI with a multivalued maximal monotone term. It is noteworthy that the author of the aforementioned work expressed the same view as we do: in his introduction, he says "Little consideration was given to the convergence of $\mathcal{A}(u_h)$ to $\mathcal{A}(u)$", with $\mathcal{A}$ a multivalued maximal mapping. Indeed the value of the control input is one of utmost importance for control engineers. Its study is one of the topic at the heart of control theory and clearly separates this field from others that study dynamical systems from other points of view.

Chapter 2

Analysis of Discrete-Time Sliding Mode Controller

Let us now get to the heart of the matter with this chapter dedicated to the study of the discrete-time SMC. First we introduce the main tool we use in our analysis: the Affine Variational Inequality framework. We also present the Linear Complementary Problem framework, which is close to the former. We try to give some motivation on why we chose to use the Affine Variational Inequality (AVI) rather than the Linear Complementary Problem (LCP). Amongst other mathematical tools we use, Convex Analysis comes second and the reader feeling the need to brushing up in that topic may for instance read Appendix A. After this laying of (mathematical) foundations, we study the discretization ECB-SMC controllers. The discretization of both parts of the control input is studied. The sliding variable is shown to be discrete-time finite-time Lyapunov stable. The perturbation rejection is studied and a concept of discrete-time sliding phase is introduced. Then the convergence of the discrete-time control input to the continuous-time one is investigated. We then switch to HOSMC with the twisting algorithm. We first try to simply discretize it using the implicit method. However, with the resulting discrete-time controller for almost all initial conditions, the state of the system ends up cycling between two values, inducing numerical chattering. Fortunately, a small modification of the controller ensures that it steers the states to the origin. With the dynamics given by the double integrator, the resulting system is globally finite-time Lyapunov stable. The Lyapunov function is inspired by the one used in [88] for the continuous-time case. Then, we touch upon two topics: discrete-time sliding mode observers and better attenuation of matched perturbations. The last subject being a trial at reducing the gap between the perfect rejection of perturbation in continuous-time versus the attenuation in discrete-time

2.1 Variational Inequalities and Complementarity Problems

We shall now quickly present some aspects of the Complementary Problem (CP) and Variational Inequality (VI) frameworks. Both have been extensively studied in the fields of Mathematical Programming and Optimization. In the following we present some of the basic concepts and results that emerged through all those investigations. We motivate also in the last part the use of VI over Complementary Problem (CP) for the discrete-time SMC.

2.1.1 Complementarity Problems

The field of complementary problems emerged as a way of unifying linear and quadratic programming problems as well as bimatrix game problems. It became an object of study of itself in the 60's in mathematical programming and has since found applications in contact mechanics among other engineering fields.

Let us first present an instance of LCP: given $M \in \mathbb{R}^{n\times n}$, $q \in \mathbb{R}^n$, the objective is to find $z \in \mathbb{R}^n$ such that

$$w = q + Mz$$
$$0 \leq z \perp w \geq 0.$$

Such problem is denoted by LCP(q, M). Let us connect the LCP with an object that the reader might be more familiar with: the KKT conditions of a QP. Consider the following quadratic program:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \tfrac{1}{2}x^T Mx + q^T x \\ \text{subject to} \quad & x \geq 0. \end{aligned}$$

The necessary conditions for optimality are given by:

$$w = M_s x + q$$
$$0 \leq w \perp x \geq 0,$$

with $M_s = (M + M^T)/2$ the symmetric part of M. In this simple example, it is easy to see that if the matrix M is symmetric (hence $M = M_s$), the LCP and the QP are tightly related. If M is positive-semidefinite, the QP is convex, and therefore solving the LCP or the QP is completely equivalent. In the case where M is not symmetric, the LCP(q, M) cannot be directly interpreted as the KKT conditions of a QP. It is still possible to construct

a QP related to an LCP(q, M): following [27, p.23], if the quadratic program

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & z^T(q + Mz) \\ \text{subject to} \quad & q + Mz \geq 0 \qquad z \geq 0. \end{aligned}$$

has a solution z^* with objective value of 0, then z^* is also a solution to LCP(q, M).

Let us now present some results on the existence and uniqueness of solution to an LCP. Those properties are given in matrix-theoretic terms that we shall now define. A good guide to the different classes of matrices and theirs relations is [26].

Definition 2.1.1 ([27, p. 147]). A matrix $M \in \mathbb{R}^{n\times n}$ is a P-matrix if one of those equivalent properties holds:

- All principal minors of M are positive: $\det M_{II} > 0$ for all $I \subseteq \{1, \ldots, n\}$
- If for all $i \in \{1, \ldots, n\}$, $x_i(Mx)_i \leq 0$, then $x = 0$.
- For all $I \subseteq \{1, \ldots, n\}$, the *real* eigenvalues of M_{II} are positive.

Theorem 2.1.2 ([27, p. 148]). *A matrix $M \in \mathbb{R}^{n\times n}$ is a* P*-matrix if and only if the* LCP(q, M) *has a unique solution for all $q \in \mathbb{R}^n$.*

Let us now investigate the relation between the well-known class of positive-definite matrices and the P-matrices.

Fact 2.1.3. A positive-definite matrix $M \in \mathbb{R}^{n\times n}$ is a P-matrix. The converse is true if M is symmetric.

Let us provide an example of a P-matrix that is not positive-definite.

Example 2.1.4 ([27, p. 147]). The matrix $M = \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix}$ is a P-matrix but is not positive definite. With $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, we have $x^T Mx = -1$. More generally, every matrix of the form

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix},$$

where A and B are P-matrices, is a P-matrix. However in general this matrix is not positive definite.

The theory of LCP goes way beyond what we have presented here. In particular the existence of solution is not guaranteed for an arbitrary LCP. If the matrix M of LCP(q, M) is not P, the existence of solution may depend on the interplay between M and q. Let us illustrate with the following instance: let M be a negative-definite matrix. If $q \leq 0$, then there is no solution to the LCP(q, M). On the other hand, if $q \geq 0$, then there is at least one (trivial) solution $z = 0$ and $w = q \geq 0$.

Before moving on to the next framework, let us state a result that has a counterpart in the case of VI. But first let us introduce another class of matrices.

Definition 2.1.5 ([27, p. 147]). A matrix $M \in \mathbb{R}^{n\times n}$ is a P_0-matrix if one of those equivalent properties holds:

- All principal minors of M are nonnegative: $\det M_{II} \geq 0$ for all $I \subseteq \{1, \dots, n\}$
- For each $z \neq 0$, there exists $k \in \{1, \dots, n\}$ such that $z_k \neq 0$ and $z_k(Mz)_k \geq 0$.
- For all $I \subseteq \{1, \dots, n\}$, the *real* eigenvalues of M_{II} are nonnegative.
- For each $\varepsilon > 0$, $M + \varepsilon I$ is a P-matrix.

Now we can characterize the cases where the solution z of the LCP(q, M) is not unique, but the vector w is.

Theorem 2.1.6. *Let $M \in \mathbb{R}^{n\times n}$. The following statements are equivalent:*

- *If the* LCP(q, M) *is solvable and $\tilde{z}$, $\hat{z}$ are any two solutions, then $M\tilde{z} = M\hat{z}$. If this is the case, then we say that the* w-uniqueness *of the LCP holds.*
- *Every vector whose sign is reversed by M belongs to the nullspace of M, that is*

$$[z_i(Mz)_i \leq 0 \text{ for all } i \in \{1, \dots, n\}] \quad \Rightarrow \quad [Mz = 0].$$

- *M is a P_0-matrix and for each $I \subseteq \{1, \dots, n\}$ with $\det M_{II} = 0$, the columns of $M_{\bullet I}$ are linearly dependent.*

This theorem gives us both a uniqueness result for w and also the structure of the solution set to LCP(q, M). Let us now switch to the other optimization framework.

2.1.2 Variational Inequalities

The variational inequality problem was first studied in the context of nonlinear partial differential operator by Guido Stampacchia and his collaborators. Those developments in an infinite-dimensional setting were paralleled in mathematical programming in finite-dimensional spaces. In this work, we only need to consider the second case. The material presented here is taken from [35]. We shall begin with the general form of a Variational Inequality (VI).

Definition 2.1.7. Given $K \subseteq \mathbb{R}^n$ and a mapping $F : K \to \mathbb{R}^n$ the *variational inequality*, denoted VI(K, F), is to find a vector $x \in K$ such that

$$\langle y - x, F(x) \rangle \geq 0, \qquad \text{for all } y \in K. \tag{2.1.1}$$

In this thesis, we only deal with sets that are bounded, closed and convex, and with continuous functions. If K is closed and convex, the VI (2.1.1) can be reformulated as a nonlinear Generalized Equation (GE)

$$0 \in F(x) + N_K(x), \tag{2.1.2}$$

with $N_K(x)$ the normal cone to K at x, introduced in Definition A.1.9. We make use of this equivalence between (2.1.1) and (2.1.2) in the analysis of the discrete-time sliding mode control. Moreover we study in particular the case where the mapping F is affine. This type of VI is called AVI and an instance is denoted as AVI(K, q, M) with $F(x) = Mx + q$. The solution set of the VI(K, F) is written as SOL(K, F).

As for the LCP, let us try to connect the VI with the KKT conditions of an optimization problem. Let us consider the following nonlinear one:

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad f(x) \\ &\text{subject to} \quad x \in K, \end{aligned} \tag{2.1.3}$$

with $f(\cdot)$ a C^1 function on an open superset of K. Since K is convex, the minimum principle in nonlinear programming implies that any minimizer x of (2.1.3) must satisfy:

$$\langle y - x, \nabla f(x) \rangle \geq 0, \quad \text{for all } y \in K. \tag{2.1.4}$$

Furthermore if f is convex, then (2.1.3) and (2.1.4) are equivalent. At this point, one natural question is whether a VI(K, F) with a convex set K and a function $F(\cdot)$ is always a stationary point of an optimization problem of the form (2.1.3). This would imply that there exists a scalar function $f(\cdot)$ such that $F(x) = \nabla f(x)$ for all $x \in K$. Symmetry is again a key part of the answer as stated in the following theorem.

Theorem 2.1.8 ([35, p. 14]). *Let $F\colon K \to \mathbb{R}^n$ be C^1 on an open subset $K \subseteq \mathbb{R}^n$. The following statements are equivalent:*

- *there exists a real-valued function $f(\cdot)$ such that $F(x) = \nabla f(x)$ for all $x \in K$.*
- *the Jacobian matrix $JF(x)$ is symmetric for all $x \in K$.*

If we look at the affine case, the problem in (2.1.3) becomes a QP if K is a polyhedron. As in the LCP case, if the matrix M of the AVI is not symmetric, then it is not possible to relate the program (2.1.3) with the AVI(K, q, M).

Let us now present results about the existence of solution to a VI.

Theorem 2.1.9 ([35, p. 148]). *Let $K \subseteq \mathbb{R}^n$ be compact convex and let $F\colon K \to \mathbb{R}^n$ be continuous. The set* SOL(K, F) *is nonempty and compact.*

This result is used in this thesis to show existence of the control input value. The uniqueness of solution to a VI is not as definite as with LCP, even for AVI. We now present some results for VI and AVI, and this topic is again discussed in Section 2.3.2.

Theorem 2.1.10. *Consider* AVI(K, q, M)*, with K a rectangle (that is $K = \{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i \, \forall i \in \{1, \dots, n\}\}$ for some vectors $a < b$). If M is a* P*-matrix, then the solution set is always a singleton.*

Proof. Using Theorem 4.3.2 p. 372 and Example 4.2.9 p. 361 in [35] yields the result. □

We shall now present a result similar in spirit to Theorem 2.1.6, but pertaining to a VI. Let us first formally define the property we want to study.

Definition 2.1.11 (F-uniqueness). A VI(K, F) is said to be *F-unique* if $F($SOL$(K, F))$ is at most a singleton. Then we say that the *F-uniqueness* of the VI holds.

Let us first introduce some characterizations of the mapping F.

Definition 2.1.12 ([35, p. 154]). Let x, y be elements of K. A mapping $F\colon K \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be

- *pseudo monotone* on K if

$$(x - y)^T F(y) \geq 0 \quad \Rightarrow \quad (x - y)^T F(x) \geq 0;$$

– *monotone* on K if

$$(F(x) - F(y))^T (x - y) \geq 0.$$

It is clear that a monotone mapping is pseudo-monotone. The uniqueness property requires an additional assumption on the map $F(\cdot)$.

Definition 2.1.13. Let x, y be elements of K. A mapping $F\colon K \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be

– *pseudo monotone plus* on K if it is pseudo monotone on K and if

$$[(x - y)^T F(y) \geq 0 \text{ and } (x - y)^T F(x) = 0] \quad \Rightarrow \quad F(x) = F(y);$$

– *monotone plus* on K if it monotone on K and if

$$(x - y)^T (F(x) - F(y)) = 0 \quad \Rightarrow \quad F(x) = F(y). \tag{2.1.5}$$

Now we can state the main uniqueness result for a VI.

Proposition 2.1.14. *Let $F\colon K \to \mathbb{R}^n$ be pseudo monotone plus on the convex set $K \subseteq \mathbb{R}^n$. The solution set* SOL(K, F) *is F-unique.*

Since we are working mainly with AVI, we specialize the aforementioned properties to the case $F(x) = q + Mx$. Firstly the monotone plus property (2.1.5) is also known as *positive semi-definite plus* (psd-plus), that is

$$x^T Mx = 0 \quad \Rightarrow \quad Mx = 0.$$

There exists a precise characterization of those matrices.

Proposition 2.1.15 ([83]). *A matrix $M \in \mathbb{R}^{n\times n}$ is psd-plus if and only if M can be decomposed into the form $E^T AE$ for some matrix $E \in \mathbb{R}^{r\times n}$ and some matrix $A \in \mathbb{R}^{r\times r}$ of the form $A = I + B$ with B skew-symmetric and r the rank of M.*

In Section 2.3.2 we shall present uniqueness result for AVI with a class of matrices larger than the psd-plus ones.

2.1.3 Choice of framework for working with the inclusion $-\lambda \in \mathrm{Sgn}(\sigma)$

We shall now see how we can express the auxiliary system in σ associated to a discrete-time SMC in those two frameworks. In the second part we give some reasons on why we found it more interesting to use VIs. In SMC, the inclusion $-\lambda \in \mathrm{Sgn}(\sigma)$ is the one modeled using the aforementioned optimization tools. Let us study the simple discrete-time system:

$$\begin{cases} \sigma = q + M\lambda & \text{(2.1.6a)} \\ -\lambda \in \mathrm{Sgn}(\sigma) & \text{(2.1.6b)} \\ \sigma, \lambda \in \mathbb{R}^p, \end{cases}$$

within the two frameworks. Let us start with the approach using an LCP: following the presentation in [81], let us split λ in positive and negative parts λ_+ and λ_-. To properly define the value at $\lambda = 0$, we impose that

$$\lambda_+ + \lambda_- = 1. \tag{2.1.7}$$

We also split σ in positive and negative parts σ_+ and σ_-. The inclusion $-\lambda \in \mathrm{Sgn}(\sigma)$ is then transformed into the complementarity relations

$$\begin{aligned} 0 \leq \sigma_+ \perp \lambda_+ \geq 0 \\ 0 \leq \sigma_- \perp \lambda_- \geq 0. \end{aligned} \tag{2.1.8}$$

Starting from the relation (2.1.6a) and using (2.1.7), we get

$$\sigma_+ - \sigma_- = q + M(2\lambda_+ - 1).$$

Coupled with the complementarity conditions (2.1.8), we get the LCP

$$\begin{bmatrix} \sigma_+ \\ \lambda_- \end{bmatrix} = \begin{bmatrix} q - M1 \\ 1 \end{bmatrix} + \begin{bmatrix} 2M & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \lambda_+ \\ \sigma_- \end{bmatrix}$$
$$0 \leq \begin{bmatrix} \sigma_+ \\ \lambda_- \end{bmatrix} \perp \begin{bmatrix} \lambda_+ \\ \sigma_- \end{bmatrix} \geq 0.$$

Let us define $M_{\mathrm{LCP}} := \begin{bmatrix} 2M & I \\ -I & 0 \end{bmatrix}$ and $q_{\mathrm{LCP}} := \begin{bmatrix} q - M1 \\ 1 \end{bmatrix}$ as well as the variables

$$z = \begin{bmatrix} \lambda_+ \\ \sigma_- \end{bmatrix} \in \mathbb{R}^{2p} \quad \text{and} \quad w = \begin{bmatrix} \sigma_+ \\ \lambda_- \end{bmatrix} \in \mathbb{R}^{2p}. \tag{2.1.9}$$

We can now write more compactly the LCP as

$$\begin{aligned} w &= q_{\mathrm{LCP}} + \mathcal{M}_{\mathrm{LCP}} z \\ 0 &\leq z \perp w \geq 0. \end{aligned} \tag{2.1.10}$$

The matrix $\mathcal{M}_{\mathrm{LCP}}$ is not a P-matrix due to the lower-right 0 block. Take any vector z with $\lambda_+ = 0$ and $\sigma_- \in \mathbb{R}^p$ nonzero. Then for all $i \in \{1, \dots, n\}$, we have $z_i(\mathcal{M}z)_i = 0$, which precludes $\mathcal{M}_{\mathrm{LCP}}$ from being a P-matrix per the second statement in Definition 2.1.1. Hence the existence and uniqueness of λ does not immediately follow from the property of the matrix $\mathcal{M}$, but rather requires more work. We postpone this analysis to Appendix B since the proposed approach requires some further results in LCP which are of no use for the rest of this thesis.

On the other hand, let us transform the system (2.1.6) into an AVI. Merging the relations (2.1.6a) and (2.1.6b) yields the Generalized Equation (GE)

$$0 \in \sigma - q + \mathcal{M}\,\mathrm{Sgn}(\sigma).$$

The inclusion $-\lambda \in \mathrm{Sgn}(\sigma)$ being equivalent to the inclusion $\sigma \in N_{[-1,1]^p}(-\lambda)$ (see Fact A.1.17), we can transform this GE into

$$0 \in q + \mathcal{M}\lambda - N_{[-1,1]^p}(-\lambda).$$

The hypercube $[-1, 1]^p$ being symmetric, we note that $-N_{[-1,1]^p}(-\lambda) = N_{[-1,1]^p}(\lambda)$ and thus we have the generalized equation

$$0 \in q + \mathcal{M}\lambda + N_{[-1,1]^p}(\lambda), \tag{2.1.11}$$

which is the equivalent form of the AVI$(K, q, \mathcal{M})$ with $K = \mathcal{B}_\infty$, the unit ball the maximum norm.

We shall now compare the two approaches. First while working with the LCP (2.1.10), we deal with vectors in $\mathbb{R}^{2p}$ while with the AVI approach, we are still in $\mathbb{R}^p$. Furthermore the vectors z and w in (2.1.9) are composite, in the sense that they comprise parts of the sliding variable and the control input. On the other hand, the GE (2.1.11) involves only the control input λ. These facts mostly pertain to the analysis of the discrete-time SMC: the existence of the control input in the AVI case follows directly from Theorem 2.1.9. The uniqueness is also characterized by the property of the matrix $\mathcal{M}$: if it is a P-matrix, then both σ and λ are unique for all $q \in \mathbb{R}^p$.

2.1.4 Discussion about Tools and Framework

A NOTE ON $PB = C^T$ We already saw the importance of symmetry to link either an LCP or a VI to a "classical" optimization problem. We shall see that this play again a role for the analysis of the discrete-time SMC. But before, let us do a small detour by the continuous-time case: consider the Differential Inclusion (DI)

$$\dot{x} + \mathcal{A}(x) \ni 0. \tag{2.1.12}$$

As mentioned in Section 1.2, this DI enjoys some nice properties, like existence and uniqueness of solution, if the multivalued mapping $\mathcal{A}$ is maximal monotone. In this case, results on the discretization of the DI (2.1.12) are also available, in particular for an implicit discretization of the DI in [96]. There it is shown that both the discretized solution x_λ and the Moreau-Yosida ([95, p. 540]) approximant $\mathcal{A}_\lambda(x_\lambda)$, which is in sliding mode the action of control input (Bu in the linear case), converge to the continuous-time ones as $\lambda \to 0$. This approach tackles directly the full system in x, as opposed to our approach which uses the auxiliary system in σ. It enables also to see that both the state, but more critically the discrete-time control input converges to the continuous time one with a full implicit discretization. In the linear case, we use a better discretization scheme since Zero-Order Hold (ZOH) integrates the dynamics exactly. This enables us to go back-and forth between the plant and the sliding dynamics. We shall state again that with an explicit discretization, the discrete-time control does not converge to the continuous time one. In the SMC context, we have on the linear case $\mathcal{A}(x) = BT(Cx)$, with $T(\cdot) = \mathrm{Sgn}(\cdot)$ a maximal monotone mapping. Suppose that there exists a change of basis matrix $R = R^T$ with $z = Rx$ such that $RB = (CR^{-1})^T$. Then by Theorem A.2.5, $\mathcal{A}'(z) := RBT(CR^{-1}z)$ enjoys the maximal monotonicity property. Therefore the new DI

$$\dot{z} + \mathcal{A}'(z) \ni 0.$$

has all the nice properties. It is then of interest to determine whether it is possible to find such a matrix R. First note the following equivalences:

$$RB = (CR^{-1})^T \quad \Leftrightarrow \quad R^T RB = C^T \quad \Leftrightarrow \quad \exists P = P^T > 0 \text{ s.t. } PB = C^T$$

For the last equivalence, the implication is direct. The reverse is true since it is always possible to find a square root of a symmetric positive-definite matrix (either by diagonalizing it using the spectral theorem and by taking the square root of the (positive) eigenvalues or we can use an SVD decomposition, which is also computationally more adequate). Hence

the problem is reduced to finding a solution to $PB = C^T$. Note that the existence of a matrix $P = P^T > 0$ solution to $PB = C^T$ is a common condition in passivity, through the celebrated Kalman-Yakubovitch-Popov Lemma, see [19]. This kind of linear matrix equality has been studied as early as the 70's [68], but it seems that it is only in [60] that we can find a suitable result. In this work, the authors study the linear matrix equation $B^T PB = B^T C^T$ with $P = P^T > 0$. It is clear that it is necessary that this problem has a solution for the equation $PB = C^T$ to also have one. Let us show that a particular solution to $B^T PB = B^T C^T$ is solution to $PB = C^T$. We restrict ourselves to the case where CB is positive-definite, which implies that $\ker B$ is reduced to a singleton. In the following, we need the SVD of $B \in \mathbb{R}^{n\times p}$, which is

$$B = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T \quad \text{with } V^T V = I_p,\ U^T U = I_n, \tag{2.1.13}$$

and $S \in \mathbb{R}^{p\times p}$ is a diagonal matrix with nonzero entries, $V \in \mathbb{R}^{p\times p}$, $U \in \mathbb{R}^{n\times n}$. Let us now state the result we need from the aforementioned reference.

Theorem 2.1.16 (Theorem 2.2 in [60]). *The linear matrix equation $B^T PB = B^T C^T$ has a symmetric positive definite solution P if and only if*

$$CB = B^T C^T, \quad CB > 0,$$

in which case the general symmetric positive definite solution is

$$P = U \begin{bmatrix} P_0 & P_{12} \\ P_{12}^T & P_{12}^T P_0^{-1} P_{12} + Z_{22} \end{bmatrix} U^T, \tag{2.1.14}$$

where $P_0 = S^{-1} V^T B^T C^T V S^{-1}$ and both $P_{12} \in \mathbb{R}^{p\times(n-p)}$ and $Z_{22} \in \mathbb{R}^{(n-p)\times(n-p)}$ are arbitrary.

Let us check that if $B^T PB = B^T C^T$ has a symmetric positive definite solution P as in (2.1.14) with $P_{12}^T = \begin{bmatrix} 0 & I_{n-p} \end{bmatrix} U^T C^T V S^{-1}$, then $PB = C^T$. Let us first transform P_0 using the SVD in (2.1.13):

$$P_0 = S^{-1} V^T V \begin{bmatrix} S & 0 \end{bmatrix} U^T C^T V S^{-1} = \begin{bmatrix} I_p & 0 \end{bmatrix} U^T C^T V S^{-1}.$$

Hence we have

$$PB = U \begin{bmatrix} \begin{bmatrix} I_p & 0 \end{bmatrix} U^T C^T V S^{-1} & P_{12} \\ & P_{12}^T \quad Z_{22} \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix} V^T$$

$$= U \begin{bmatrix} \begin{bmatrix} I_p & 0 \end{bmatrix} U^T C^T V \\ \begin{bmatrix} 0 & I_{n-p} \end{bmatrix} U^T C^T V \end{bmatrix} V^T = C^T.$$

The linear matrix equation $PB = C^T$ has a solution $P = P^T > 0$ if and only if CB is *symmetric* positive definite. Once again the symmetry plays an important role in characterizing a system. In contrast, the results in the following section do not require this property.

2.2 ECB-SMC

2.2.1 Discrete-time sliding mode controllers

Discretization methods for discrete-time controllers

Let us now tackle the construction of a discrete-time sliding mode controller. First let us recall the definition of the multivalued signum function.

Definition 2.2.1 (Multivalued signum function). Let $x \in \mathbb{R}$. The multivalued signum function $\mathrm{Sgn} : \mathbb{R} \rightrightarrows \mathbb{R}$ is defined as:

$$\mathrm{Sgn}(x) = \begin{cases} \{1\} & x > 0 \\ \{-1\} & x < 0 \\ [-1, 1] & x = 0. \end{cases} \tag{2.2.1}$$

If $x \in \mathbb{R}^n$, then the multivalued signum function $\mathrm{Sgn} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is defined as: for all $j = 1, \ldots, n$, $(\mathrm{Sgn}(x))_j := \mathrm{Sgn}(x_j)$.

The first step is to transform the continuous-time system (1.1.1) in a discrete-time one. Using the ZOH scheme, we obtain:

$$x_{k+1} = e^{Ah} x_k + B^* u_k^{eq} + B^* u_k^{s}, \tag{2.2.2}$$

with

$$B^* := \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} B \mathrm{d}\tau. \tag{2.2.3}$$

From now on, u^{eq} and u^s are sampled control laws defined as right-continuous step functions:

$$u^{eq}(t) = u_k^{eq} \quad t \in [t_k, t_{k+1})$$

$$u^s(t) = u^s_k \qquad t \in [t_k, t_{k+1}).$$

The goal of the discretization process is to choose the sequences $\{u^{eq}_k\}$ and $\{u^s_k\}$ such that the discrete-time closed-loop system exhibits properties as close as possible to the ones with a continuous-time set-valued controller. Let us introduce some possible definitions of u^{eq}_k and u^s_k obtained using classical discretization method. For the equivalent part u^{eq} we have

$$u^{eq}_{k,e} = -(CB)^{-1}CAx_k \qquad \text{explicit input,} \tag{2.2.4a}$$
$$u^{eq}_{k,i} = -(CB)^{-1}CAx_{k+1} \qquad \text{implicit input,} \tag{2.2.4b}$$
$$u^{eq}_{k,m} = 1/2(u^{eq}_{k,e} + u^{eq}_{k,i}) \qquad \text{midpoint input,} \tag{2.2.4c}$$

and the two possibilities for the discontinuous control u^s_k are

$$u^s_k = -\alpha \operatorname{sgn}(\sigma_k) \qquad \text{explicit input,} \tag{2.2.5a}$$
$$u^s_k \in -\alpha \operatorname{Sgn}(\sigma_{k+1}) \qquad \text{implicit input.} \tag{2.2.5b}$$

Suppose that the implicit discretization (2.2.5b), introduced in [4] and [5], is used to discretize the discontinuous part of the controller. Both the analysis and the computation of the control law involve an auxiliary square subsystem combining the dynamics of the sliding variable and the nonsmooth control law:

$$\begin{cases} \widetilde{\sigma}_{k+1} = \sigma_k + CB^* u^s_k \\ -u^s_k \in \alpha \operatorname{Sgn}(\widetilde{\sigma}_{k+1}), \end{cases} \tag{2.2.6}$$

with two unknowns: $\widetilde{\sigma}_{k+1}$ and u^s_k. This system can be seen as the is the discrete counterpart of (1.1.3). The use of $\widetilde{\sigma}_{k+1}$ instead of σ_{k+1} is deliberate in order to highlight that in general the two quantities are different even in the nominal case: the state dynamics is given by (2.2.2) with u^{eq}_k usually given by the discretization of its continuous-time version (1.1.2). Hence nothing warrants that $C(e^{Ah}x_k + B^* u^{eq}_k) = \sigma_k$. We take care of this design issue in the last part of this section. But before let us provide some details about the computation of the control law in the most generic setting.

Definition and properties of the implicitly discretized discontinuous control input

We analyze the system (2.2.6) using the AVI formalism introduced at the beginning of this chapter. Let $N_{[-\alpha,\alpha]^p}(\lambda)$ be the normal cone to the box $[-\alpha, \alpha]^p$ at λ, as defined in

Definition A.1.9. Using Fact A.1.17, we have the equivalence

$$-u_k^s \in \mathrm{Sgn}(\widetilde{\sigma}_{k+1}) \Longleftrightarrow \widetilde{\sigma}_{k+1} \in N_{[-\alpha,\alpha]^p}(-u_k^s), \tag{2.2.7}$$

which enables us to transform (2.2.6) into the generalized equation:

$$0 \in \sigma_k + CB^* u_k^s + N_{[-\alpha,\alpha]^p}(u_k^s). \tag{2.2.8}$$

The inclusion (2.2.8) is satisfied if and only if u_k^s is the solution of the AVI: Find $z \in [-\alpha, \alpha]^p$ such that

$$\langle (y - z), (\sigma_k + CB^* z) \rangle \geq 0, \quad \forall y \in [-\alpha, \alpha]^p. \tag{2.2.9}$$

Let $\mathrm{SOL}(CB^*, Cx_k)$ denote the set of all solutions to the AVI (2.2.9). Its well-posedness is studied as follows:

Lemma 2.2.2. *The AVI* (2.2.9) *has always a solution.*

Proof. Since the mapping $z \mapsto CB^* z + \sigma_k$ is continuous and the control set $[\alpha, \alpha]^p$ is compact, we can apply Theorem 2.1.9. □

Lemma 2.2.3. *The AVI* (2.2.9) *has a unique solution for all* $\sigma_k \in \mathbb{R}^p$ *if and only if* CB^* *is a* P*-matrix.*

Proof. This is given by Theorem 2.1.10. □

In most ECB-SMC systems, $CB^* > 0$, and therefore is a P-matrix. This AVI-based approach enables us to analyze a larger class of systems, compared to previous approaches where it is supposed that CB^* is scalar as in [49]. The solution u_k^s is a function of σ_k (hence x_k) and if CB^* is a P-matrix, the solution map $Cx_k \mapsto u_k^s = \mathrm{SOL}(CB^*, \sigma_k)$ is Lipschitz continuous. It is also noteworthy that the controller is non-anticipative. Solving the AVI (2.2.9) is the topic of Section 3.1.

Definition 2.2.4 (discrete-time sliding phase). When u_k^s is in the interior of $[-\alpha, \alpha]^p$, we say that the closed-loop system is in the *discrete-time sliding phase*. The inclusion $-\widetilde{\sigma}_{k+1} \in N_{[-\alpha,\alpha]^p}(u_k^s)$ implies that in this case the normal cone is reduced to the singleton $\{0\}$. Thus the sliding variable $\widetilde{\sigma}_k$ is zero in the discrete-time sliding phase (however in general σ_k does not vanish).

Now that we have discussed the existence and uniqueness properties of solutions to (2.2.6), we need to tackle the computation of the equivalent part of the control.

Exact discrete equivalent control

Let us complete this sliding mode control scheme for a discrete-time LTI plant by providing the equivalent part of the control. As recalled in (1.1.3), in continuous time, u^{eq} is defined such that the dynamics of the sliding variable depends only on the input u^s. Mimicking the continuous-time design, we start from (2.2.2) and by multiplying by C, we obtain in discrete-time the relation

$$\sigma_{k+1} = Cx_{k+1} = Ce^{Ah}x_k + CB^* u_k^{eq} + CB^* u_k^s. \tag{2.2.10}$$

Our design objective is to make σ_{k+1} depend only on σ_k and u^s. Hence we want that $Ce^{Ah}x_k + CB^* u_k^{eq} = \sigma_k$, which gives

$$u_k^{eq} = (CB^*)^{-1} C(I - e^{Ah})x_k. \tag{2.2.11}$$

If we substitute this expression for u_k^{eq} in (2.2.10), then, as expected, we obtain

$$\sigma_{k+1} = \sigma_k + CB^* u_k^s.$$

Using the implicitly discretized discontinuous part of the control $-u_k^s \in \alpha \operatorname{Sgn}(\sigma_{k+1})$, the discrete-time sliding variable dynamics is

$$\begin{cases} \sigma_{k+1} = \sigma_k + CB^* u_k^s \\ -u_k^s \in \alpha \operatorname{Sgn}(\sigma_{k+1}), \end{cases} \tag{2.2.12}$$

which is the discrete counterpart of (1.1.3). This system has the same structure as in (2.2.6), although with the important difference that $\widetilde{\sigma}_{k+1} = \sigma_{k+1}$. This system is studied in the next section, where the finite-time stability and robustness is analyzed. Let us first state the following result.

Lemma 2.2.5. *If CB^* is a* P*-matrix, then the only equilibrium pair of the system* (2.2.12) *is* $(\sigma_*, u_*^s) = (0, 0)$.

Proof. A pair (σ, u^s) is an equilibrium of (2.2.12) if and only if $CB^* u^s = 0$. If CB^* is a P-matrix, then it has full-rank and $CB^* u^s = 0$ is equivalent to $u^s = 0$. From the definition of the $\operatorname{Sgn}(\cdot)$ multifunction in (2.2.1), this is only possible if $\sigma = 0$. □

To sum up, with the proposed scheme the two control inputs are

$$\begin{cases} u_k^{eq} = (CB^*)^{-1} C(I - e^{Ah})x_k \\ u_k^s \qquad \text{solution of (2.2.12).} \end{cases} \tag{2.2.13}$$

This controller is nonanticipative since u_k^{eq} depends only on the model parameters and x_k. Moreover u_k^s is the unique solution to (2.2.12) given that $CB^* > 0$, using similar arguments as in Lemma 2.2.3. This controller retains the structure of the continuous-time sliding mode controller. It is different from the approach that can be found in [108] or [79] since in our case, the equivalent part u^{eq} is not chosen as the solution to a deadbeat control problem. As a result, the magnitude of the control input in (2.2.12) is $O(1)$ with respect to the sampling period h, whereas it is $O(h^{-1})$ in the deadbeat case, see [79]. In [42], this expression for the equivalent control u^{eq} was already derived, when the sliding variable is scalar.

2.2.2 Stability and convergence properties

We shall now investigate the stability of the auxiliary system (2.2.12) and some convergence properties of the control input u^s when the sampling period tends to 0. In the nominal case we are able to prove convergence of the sliding variable to 0, using Lyapunov technique, under some structural conditions that match closely the ones for the continuous-time sliding mode controller. In the case where the dynamics include matched perturbations, we show how the proposed controller attenuates their effects and that if the controller action is large enough, the system remains in a neighborhood of the sliding manifold. We study the convergence of the control input since for control theorist it is a crucial variable and it received little attention in discretization studies of differential inclusions, as mentioned in Section 1.4.4. Those results also underline the difference between the implicit and explicit discretization methods. Every property shown for the "auxiliary" system (2.2.12) remains valid for the closed-loop system (2.2.2) and (2.2.13), thanks to the structure of the controller introduced in the previous section.

Stability in the nominal case

Let us start by considering the stability of the system (2.2.12). Note that the mapping $\mathrm{Sgn}(\cdot)$, as introduced in Definition 2.2.1, is maximal monotone. This concept is introduced in Appendix A.2. This property, as well as another important one, is given by:

$$\langle v_1 - v_2, x_1 - x_2 \rangle \geq 0, \ \forall v_i \in \mathrm{Sgn}(x_i), \ i = 1, 2 \tag{2.2.14}$$

$$0 \in \mathrm{Sgn}(x) \iff x = 0. \tag{2.2.15}$$

The positive-definitiveness property of CB^* is pivotal to the results presented in this section. Even if it is not explicit with the current notations, CB^* depends on the sampling period h. The following lemma gives some insight of when this condition is fulfilled.

Lemma 2.2.6. *Suppose that CB is positive definite and B^* is the matrix given by the ZOH integration of B as in* (2.2.3)*. There exists an interval $I = (0, h^*] \subset \mathbb{R}_+$, $h^* > 0$, such that if the sampling period $h \in I$, then CB^*/h and CB^* are positive definite.*

Proof. Let $h > 0$, CB_s and CB_s^* be the symmetric parts of CB and CB^*, respectively. Let $\Delta \coloneqq CB_s^*/h - CB_s = \sum_{l=1}^{\infty} \frac{CA^lB + B^T(A^l)^TC^T}{2(l+1)!} h^l = O(h)$. Since CB_s is symmetric, it is also normal. Hence we can apply Corollary D.3, which yields that for any eigenvalue μ of CB_s^*/h, $\min_{\lambda} |\lambda - \mu| \leq \|\Delta\|$, with λ an eigenvalue of CB_s and $\|\cdot\|$ the spectral norm. By definition, Δ is a symmetric matrix with real entries. Hence $\|\Delta\| = \delta_{max}$, the largest module of any eigenvalue of Δ. Let $\gamma > 0$ be the smallest eigenvalue of CB_s. If $\delta_{max} < \gamma$, then every eigenvalue of CB_s^*/h is positive and since CB_s^*/h is by definition symmetric, CB_s^*/h is positive definite. It is easy to see that $\Delta \to 0$ as $h \to 0$ and that Δ depends continuously on h. Therefore by Corollary D.2, the eigenvalues of Δ are continuous functions of h. Then it is always possible to find h^* such that $\delta_{max} < \gamma$ for all $0 < h < h^*$, which implies that CB_s^*/h is positive definite as well as CB_s^*. □

Lemma 2.2.7. *If CB^* is symmetric positive definite, then the equilibrium state $\sigma^* = 0$ of* (2.2.12) *is globally Lyapunov stable with a quadratic Lyapunov function.*

Proof. Let $V(\sigma_k) \coloneqq \sigma_k^T P \sigma_k$ with $P = (CB^*)^{-1}$ be a candidate Lyapunov function. Along the trajectories of the system (2.2.12), one obtains:

$$\begin{aligned} V(\sigma_{k+1}) - V(\sigma_k) &= \sigma_{k+1}^T P \sigma_{k+1} - \sigma_k^T P \sigma_k \\ &= \sigma_{k+1}^T P \sigma_{k+1} - (\sigma_{k+1} - CB^* u_k^s)^T P (\sigma_{k+1} - CB^* u_k^s) \\ &= -(u_k^s)^T CB^* u_k^s + 2(u_k^s)^T \sigma_{k+1}. \end{aligned}$$

Using (2.2.14) with $v_1 = u_k^s$, $v_2 = 0$, $x_1 = \sigma_{k+1}$, and $x_2 = 0$ yields $(u_k^s)^T \sigma_{k+1} \leq 0$. Since CB^* is positive definite, the first term is always nonpositive and vanishes if and only if $u_k = 0$. Using (2.2.15) we then infer that $\sigma_{k+1} = 0$ and (2.2.12) gives us that $\sigma_k = 0$. This completes the proof. □

We can also use a non-quadratic Lyapunov function, inspired by the one presented in [107] in the continuous-time case. As we shall see, it relaxes the symmetry condition on the matrix CB^*.

Lemma 2.2.8. *If CB^* is positive definite, then the equilibrium state $\sigma^* = 0$ of* (2.2.12) *is globally Lyapunov stable.*

Proof. Let $V(\sigma_k) := \|\sigma_k\|_1 = -(u^s_{k-1})^T\sigma_k$ be the candidate Lyapunov function, and $-u^s_{k-1} \in \alpha\,\mathrm{Sgn}(\sigma_k)$. The function V is positive definite, radially unbounded, and decrescent since $-(u^s_{k-1})^T\sigma_k = \alpha\|\sigma_k\|_1^2$ and $\alpha > 0$. Let us study the variations of V:

$$\begin{aligned} V(\sigma_{k+1}) - V(\sigma_k) &= -(u^s_k)^T\sigma_{k+1} + (u^s_{k-1})^T\sigma_k \\ &= -(u^s_k)^T(\sigma_k + CB^*u^s_k) + (u^s_{k-1})^T\sigma_k \\ &= -(u^s_k)^T CB^*u^s_k + \langle u^s_{k-1} - u^s_k, \sigma_k\rangle. \end{aligned} \tag{2.2.16}$$

The first term is always nonpositive with the hypothesis on CB^*. For the second term, let us recall that from the relation (2.2.7), we have $\sigma_k \in N_{[-\alpha,\alpha]^p}(-u_{k-1})$. From the definition of the normal cone, we know that the second term is always nonpositive. The proof is then completed in a similar fashion to the previous one. □

Proposition 2.2.9. *If the hypothesis of either Lemma 2.2.7 or 2.2.8 are satisfied, then the fixed point* $(\sigma, u) = (0, 0)$ *of* (2.2.12) *is globally finite-time Lyapunov stable.*

Proof. In each case, the difference $V(\sigma_{k+1}) - V(\sigma_k)$ consists of $-(u^s_k)^T CB^*u^s_k$ plus a nonpositive term. Since CB^* is positive definite, it holds that $-(u^s_k)^T CB^*u^s_k \le -\beta\|u^s_k\|^2$, with $\beta > 0$ the smallest eigenvalue of CB^*_s. Note that if $\sigma_{k+1} \neq 0$, then $\|u^s_k\| \ge \alpha$ and $V(\sigma_{k+1}) - V(\sigma_k) \le -\alpha^2\beta$. Iterating, one obtains $V(\sigma_{k+1}) - V(\sigma_0) \le -k\alpha^2\beta$. Let $k_0 := \lceil V(\sigma_0)/\beta\alpha^2\rceil$ and suppose that $V(\sigma_{k_0+1}) \neq 0$. Then $V(\sigma_{k_0+1}) - V(\sigma_0) \le -k_0\alpha^2\beta \le -V(\sigma_0)$. This yields $V(\sigma_{k_0+1}) \le 0$, which implies $V(\sigma_{k_0+1}) = 0$. Hence $\sigma_{k_0+1} = 0$ and $\sigma_k = 0$ for all $k > k_0$. □

To the best of our knowledge, these proofs of Lyapunov stability in the discrete-time case are new and have never beem done before for discrete-time SMC.

Stability in the perturbed case

Let us now consider the case when a matched perturbation is acting on the system. The evolution of the sliding variable σ_k is governed by

$$\sigma_{k+1} = \sigma_k + CB^*u^s_k + Cp_k, \tag{2.2.17}$$

where

$$p_k := \int_{t_k}^{t_{k+1}} e^{\mathcal{A}(t_{k+1}-\tau)}B\xi(\tau)\mathrm{d}\tau \tag{2.2.18}$$

and u_k^s is the unique solution of the generalized equation (2.2.6). Although the system neither reaches nor stays on the sliding manifold as in the continuous-time case, we shall see that it enters the discrete-time sliding phase (see Definition 2.2.4) and stays in it. Let us first present a technical lemma.

Lemma 2.2.10. *Let $M \in \mathbb{R}^{n\times n}$ and $M_s := 1/2(M + M^T)$. Suppose M is positive definite and let $\beta > 0$ be the smallest eigenvalue of M_s. Then for all $x \in \mathbb{R}^n$, $\|M^{-1}x\| \leq \beta^{-1}\|x\|$.*

Proof. M^{-1} exists since M_s is positive definite. Let ν_{min} (respectively ν_{max}) be the smallest (respectively largest) singular values of M (resp. M^{-1}). Two relations hold: $\nu_{max} = \nu_{min}^{-1}$, see Fact D.4, and $\nu_{min} \geq \beta > 0$ from Corollary D.5. Then using the spectral norm definition, we have $\|M^{-1}x\| \leq \|M^{-1}\|\|x\| = \nu_{max}\|x\| \leq \beta^{-1}\|x\|$. □

From now on, let $CB_s^* := 1/2(CB^* + (CB^*)^T)$ and let β be its smallest eigenvalue.

Proposition 2.2.11. *Suppose that CB^* is positive definite. If the controller gain $\alpha > 0$ is such that for all $k \in \mathbb{N}$, $\|Cp_k\| < \alpha\beta$, then the perturbed closed-loop system* (2.2.6) *and* (2.2.17) *enters the discrete-time sliding phase in finite time and stays in it with $\sigma_{k+1} = Cp_k$. Furthermore if $h \in (0, h^*]$, as defined in Lemma 2.2.6, then there exists an upper bound T^* on the duration of the reaching phase.*

Proof. Let $V(\sigma_k) := -u_{k-1}^{sT}\sigma_k$ with $-u_{k-1}^s \in \alpha\,\mathrm{Sgn}(\sigma_k)$. Assume that the system is initialized outside the discrete-time sliding phase. It follows that $\|u_k^s\| \geq \alpha$. Starting from (2.2.16), doing as in the proof of Proposition 2.2.9 and adding the contribution of the perturbation, we have $V(\sigma_{k+1}) - V(\sigma_k) \leq -\beta\|u_k^s\|^2 - (u_k^s)^T Cp_k$. Using the Cauchy-Schwarz inequality, we obtain $|(u_k^s)^T Cp_k| \leq \|u_k^s\|\|Cp_k\|$. To ensure that $V(\cdot)$ decreases strictly, we need $\|Cp_k\| < \beta\|u_k^s\|$. This condition is satisfied using the hypothesis on the gain α and the fact that $\beta > 0$. Note that even in the case with multiple switching surfaces, $V(\cdot)$ decreases as long as the system is not "sliding" on the intersection of all the manifolds: $\|u_k^s\| < \alpha$ can only hold when $\widetilde{\sigma}_{k+1} = 0$. If $\widetilde{\sigma}_{k+1} = 0$, then we enter the discrete-time sliding phase.

The finite-time property is derived as in the proof of Proposition 2.2.9. Let $\varkappa = \alpha\beta - \|Cp_k\|$. By assumption, $\varkappa > 0$ holds, therefore in the reaching phase, $V(\cdot)$ decreases by at least $\varkappa\alpha$ at each sampling period. Hence, $V(\sigma_k)$ converges to 0 in finite-time. Now if the system is in the discrete-time sliding phase at the time instant t_k, then $\widetilde{\sigma}_{k+1} = 0$ and $\sigma_{k+1} = Cp_k$. At time t_{k+1}, we have $\widetilde{\sigma}_{k+2} = Cp_k + CB^* u_{k+1}^s$. Let us show that $u_{k+1}^s = -(CB^*)^{-1}Cp_k$ is the unique solution to the generalized equation (2.2.6).

With this value, we have $\widetilde{\sigma}_{k+2} = 0$. Using Lemma 2.2.10 and the hypothesis, we infer $\|u^s_{k+1}\| \leq \beta^{-1}\|Cp_k\| < \alpha$. Relations between norms yield $\|u^s_{k+1}\|_\infty < \alpha$. Then $u^s_{k+1} \in (\alpha, \alpha)^p \subset \alpha\,\mathrm{Sgn}(0)$ and u^s_{k+1} is a solution to (2.2.6). With the hypothesis of the proposition, CB^* is also a P-matrix. Then Lemma 2.2.3 can be applied and yields the uniqueness property. Thus $u^s_{k+1} = -(CB^*)^{-1}Cp_k$ is the unique solution to (2.2.6) at time t_{k+1} and by induction the system stays in the discrete-time sliding phase.

In the following, we suppose that $h \in (0, h^*]$. Let $\delta^*_{max} = \|\Delta\| = \|CB^*_s/h^* - CB_s\|$ (from the proof of Lemma 2.2.6) with $h = h^*$. From the expression of k_0 in the proof of Proposition 2.2.9, the duration of the reaching phase is $hk_0 < \dfrac{V(\sigma_0)h}{\beta\alpha^2} + h$. Note that β/h is an eigenvalue of CB^*_s/h. Applying again Corollary D.3, we have $\lambda - \beta/h \leq \delta_{max} \leq \delta^*_{max}$, with λ an eigenvalue of CB_s. This yields $\gamma - \delta^*_{max} \leq \lambda - \delta^*_{max} \leq \beta/h$. Using this in the previous inequality, we get $hk_0 < \dfrac{V(\sigma_0)}{\alpha^2(\gamma - \delta^*_{max})} + h^* =: T^*$. □

Remark 2.2.12. In continuous time, the usual condition is $\alpha > \|\xi\|_{\infty,\mathbb{R}_+}$. If the perturbation ξ is continuous, then it is still possible to link this condition to the one used in the previous theorem, $\|Cp_k\| < \alpha\beta$ for all $k \in \mathbb{N}$. Using the mean value theorem for integration, we get $Cp_k = hCe^{A(t_{k+1}-t')}B\xi(t') = hCB\xi(t') + O(h^2)$, with $t' \in [t_k, t_{k+1}]$. Hence the first-order estimation for $\|Cp_k\|$ is $h\|CB\|\|\xi\| \leq h\sqrt{p}\|CB\|\|\xi\|_{\infty,\mathbb{R}_+}$. From the proof of Lemma 2.2.6, it follows that $\beta = h\lambda_{\min}(CB) + O(h^2)$, and from Corollary D.5 we have $\|CB\| > \lambda_{\max}(CB)$. For h small enough, we infer $\|CB\|\sqrt{p}/\beta \geq 1$ and therefore $\|Cp_k\| < \alpha\beta$ implies $\alpha > \|\xi\|_{\infty,\mathbb{R}_+}$. If the sliding variable is a scalar, then the converse is also true at the limit.

In the classical literature on discrete-time sliding mode, where the explicit discretization (2.2.5a) is used as for instance in [84, 97, 42], two conditions related to the sliding variable emerged: $(\sigma_{k+1} - \sigma_k)_i(\sigma_k)_i < 0$ for all $i = 1, \ldots, n$, which is necessary; and the second one is $|(\sigma_{k+1})_i| < |(\sigma_k)_i|$. With our approach the conditions for linear systems, stated in Lemmas 2.2.7 and 2.2.8 as well as Proposition 2.2.11, are on the system parameters and not on the evolution on the sliding variable, which derives from the dynamics. This is much closer to the stability results obtained in continuous-time [107].

Looking at the value of the sliding variable in the discrete-time sliding phase, $\sigma_k = Cp_{k-1}$ implies by the definition of the right-hand side that $\|\sigma_k\|_\infty$ has an upper bound proportional to $h\|\xi\|_\infty$, see Remark 2.2.12. Let us now study the influence of the gain on the control input.

Corollary 2.2.13. *Suppose that α is such that for all $k \in \mathbb{N}$, $\|Cp_k\| \leq \alpha\beta$. Then even if the controller gain is increased to $\alpha' > \alpha$, the control input u^s does not change in the discrete-time sliding phase.*

Proof. From the proof of Prop. 2.2.11, we have that u^s_k is uniquely defined as the solution to (2.2.6), and is equal to $-(CB^*)^{-1}Cp_k$ which does not depend on the controller gain. □

This is a major difference with the explicitly discretized controller, where a change in the controller gain always influences the control input. This result is also similar to the continuous-time case: within Filippov's framework, when the system is in the sliding phase, the control input is a selection of the set-valued right-hand side which does not depend on the gain, given that the latter is large enough to dominate the perturbation. In simulations (Section 3.4) and in experimental results (Chapter 4), those property have been verified.

Remark 2.2.14. Let us highlight two similarities between the continuous-time sliding mode control and the discrete-time we present here: the first is the expression of the control input value during the sliding phase. In continuous-time, with Filippov's notion of solution, we have $u^s_{\text{cont}}(t) = -\xi(t)$. In other words, the control input is the selection of the set-valued right-hand side which exactly compensates for the disturbance. With the implicit (discrete-time) controller, we have $u^s_{k+1} = -(CB^*)^{-1}Cp_k$ and p_k is an integral term involving ξ, remember (2.2.18). The connection between the two quantities is the topic of the next section. The other point is the existence of an upper bound on the reaching phase, which is denoted by T^*, as in continuous-time.

Convergence of the control input

Let us now turn our attention to the relationship between the continuous time input u_{cont} and u. In particular, we study the convergence of u to u_{cont} during the discrete-time sliding phase, which is established after $T^* < +\infty$. To the best of our knowledge, the only convergence study of this type is in [96], with a slightly different approach but which requires the symmetry of CB. We introduce the following notation: let $w\colon \mathbb{R} \to \mathbb{R}^r$ and S be any interval in $\mathbb{R}$, $\|w\|_{\infty,S} = \max_i \operatorname{ess\,sup}_{t\in S} |w_i(t)|$.

Proposition 2.2.15. *Consider the discrete-time closed-loop system given by* (2.2.17) *and* (2.2.6). *Let $\{h_n\}_{n\in\mathbb{N}}$ be any strictly decreasing sequence of positive numbers converging to* 0 *and with $h_0 < h^*$ (see Lemma 2.2.6). Suppose that the perturbation $\xi\colon \mathbb{R} \to \mathbb{R}^p$ is uniformly*

continuous, that CB is positive definite and that $\alpha > 0$ *is chosen such that the conditions of Proposition 2.2.11 are satisfied for each sampling period* h_n*. Then for any interval* $S \subseteq [T^*, \infty)$, $\lim_{h_n \to 0} \|u^s - u^s_{\text{cont}}\|_{\infty,S} = 0$.

The proof is in Section 2.2.4. Let us recall that this convergence result does not hold with an explicit discretization and that for control theorist, the control input is a quantity of paramount importance. Also this result underlines that with an implicit method, we "approximately observe" the perturbation ξ, in a consistent fashion. We expand on this in Section 2.5.

It is also interesting to study the convergence of the variation of the control variable, which we introduced in Section 1.3 as a measure of the control input chattering.

Proposition 2.2.16. *Suppose that CB is positive definite, and* ξ *is a real-valued continuously differentiable with bounded derivative function. Let* $\{h_n\}_{n\in\mathbb{N}}$ *be any strictly decreasing sequence of positive numbers converging to* 0 *with* $h_0 < h^*$*. Let* α *be chosen such that the conditions of Proposition 2.2.11 are verified for each* h_n*. Let* $T > T^*$ *with* T^* *defined in Proposition 2.2.11. Then* $\lim_{h_n \to 0} \text{Var}^T_{T^*}(u^s) = \text{Var}^T_{T^*}(u^s_{\text{cont}})$.

The proof is in Section 2.2.4. In order to compare with the control input given by the explicit discretization as in (2.2.5a) below, let us recall the conclusion from [47] and [93], valid for a 2-D linear system with an explicit discretization of the sgn function: if the sampling period is small enough, once the closed-loop system is close to the sliding manifold, it spends at most 2 consecutive sampling period on each side of the sliding surface. The control input variation is easy to compute as the sgn function is equal to +1 or −1: each time the sliding manifold is crossed, the variation increases by 2. It is then easy to see that for a small enough sampling period, the variation of the explicitly discretized control grows linearly with the inverse of the sampling period, hence it explodes as $h \to 0$.

In Sections 4.1.2, 4.2.2 and 3.4, we illustrate some results presented in this section: the similarity between the perturbation and the discrete-time control input (Proposition 2.2.11), alongside the convergence of the discrete-time control input to the continuous-time one (Proposition 2.2.15).

2.2.3 Discretization performance

The aim of this section is to analyze the performances of the controllers presented in Section 2.2.1. We reproduce here their expressions: for the equivalent part u^{eq} we have

$$\begin{aligned} u^{eq}_{k,e} &= -(CB)^{-1}CAx_k && \text{explicit input,} \\ u^{eq}_{k,i} &= -(CB)^{-1}CAx_{k+1} && \text{implicit input,} \\ u^{eq}_{k,m} &= 1/2(u^{eq}_{k,e} + u^{eq}_{k,i}) && \text{midpoint input,} \end{aligned}$$

and the two possibilities for the discontinuous control u^s_k are

$$\begin{aligned} u^s_k &= -\alpha \operatorname{sgn}(\sigma_k) && \text{explicit input,} \\ u^s_k &\in -\alpha \operatorname{Sgn}(\sigma_{k+1}) && \text{implicit input.} \end{aligned}$$

We shall here provide results on the asymptotic behavior (as $h \to 0$) for both the equivalent and the discontinuous parts of the control. In the last part, we discuss the case when there is a matched perturbation.

Discretization of the state-continuous control

Let us begin with the discretization error on u^{eq} and more specifically on its effect on the sliding variable. In other words we analyze how the invariance property in (1.1.2) is preserved after discretization. In the following, u^s is set to 0 as we want to evaluate how the equivalent part of the control manages to keep the value of the sliding variable. To this effect, let $\Delta\sigma_k := \sigma_{k+1} - \sigma_k$ be the local variation of the sliding variable due to the discretization error on u^{eq}.

With an explicit discretization of u^{eq} as in (2.2.4a), and using (2.2.2) the closed-loop discrete-time system dynamics is

$$x_{k+1} = \Phi^e_k x_k, \tag{2.2.21}$$

with $\Psi := \displaystyle\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}\mathrm{d}\tau = \sum_{l=0}^{\infty} \frac{A^l h^{l+1}}{(l+1)!}$, $\Phi^e_k := e^{Ah} - \Psi\Pi_B A$ and $\Pi_B := B(CB)^{-1}C = I - \Pi_{\ker C}$, where Π is the projection operator. In the implicit case, the recurrence equation (2.2.2) combined with (2.2.4b) yields

$$x_{k+1} = e^{Ah}x_k - \Psi\Pi_B A x_{k+1}, \tag{2.2.22}$$

that is:

$$x_{k+1} = W^{-1}e^{Ah}x_k,$$

with $W = I + \Psi\Pi_B A$. Starting from (2.2.4b), the control input value is

$$u_{k,i}^{eq} = -(CB)^{-1}CAW^{-1}e^{Ah}x_k,$$

which is non-anticipative.

Lemma 2.2.17. *Let $u^s \equiv 0$. With either an explicit or implicit discretization of u^{eq}, the discretization error $\Delta\sigma_k$ is $O(h^2)$. If the midpoint method* (2.2.4c) *is used, then the error is $O(h^3)$.*

Proof. Let us start with the implicit case. There exists a Taylor expansion for W^{-1} if $\Psi\Pi_B A$ has all its eigenvalues in the unit disk. Since $\Psi \to 0$ as $h \to 0$, it is always possible to find an h_0 such that this condition holds for all $h_0 > h > 0$. Since we are interested in an asymptotic property, such restriction on h does not play any role. The finite expansion of $W^{-1}e^{Ah}$ is:

$$W^{-1}e^{Ah} = I - (\Pi_B A - A)h + \left(\frac{\Pi_B A \Pi_B A}{2} - \Pi_B A^2 + \frac{A^2}{2}\right)h^2 + O(h^3).$$

The variation of the sliding variable is

$$\begin{aligned}\Delta\sigma_k &= \sigma_{k+1} - \sigma_k = C(W^{-1}e^{Ah} - I)x_k \\ &= h(-A + A)x_k + h^2\left(\frac{CA\Pi_B A}{2} - \frac{CA^2}{2}\right)x_k + O(h^3) \\ &= \frac{h^2}{2}CA(I - \Pi_B)Ax_k + O(h^3) = -\frac{h^2}{2}CA\Pi_{\ker C}Ax_k + O(h^3). \quad (2.2.23)\end{aligned}$$

For the explicit case, expanding the exponential and Ψ terms yields

$$\Delta\sigma_k = \frac{h^2}{2}CA\Pi_{\ker C}Ax_k + O(h^3). \quad (2.2.24)$$

With the midpoint method (2.2.4c), the recurrence equation is

$$x_{k+1} = 0.5(e^{Ah}x_k - \Psi\Pi_B Ax_k) + 0.5(e^{Ah}x_k - \Psi\Pi_B Ax_{k+1}).$$

The last two terms are the right-hand sides of (2.2.21) and (2.2.22). Thus the contribution in h^2 are given in (2.2.24) and (2.2.23). Hence the term in h^2 is canceled and the error is $O(h^3)$. □

Discretization of both control inputs

In the following, we consider the sliding variable dynamics with the state-continuous and discontinuous control. It is expected that σ goes to 0 and once it reaches zero, stays at this value. The proposed metric to measure the performance of the discrete-time controller is the Euclidean norm of the sliding variable when the system state is close to the sliding manifold. Let $\varepsilon_k := \|\sigma_{k+1}\|$ be the discretization error when $\|\sigma_k\|$ is small enough. We consider also the case with a matching perturbation (which is $O(h)$), leading to dynamics as in (2.2.17).

EXPLICIT DISCRETIZATION In the sliding mode literature, several proposals (seven of them are listed in [44]) have been made to analyze the behavior of the closed-loop system near the sliding manifold and to propose new variable structure control strategies. Despite this, it is still difficult to analyze the behavior near the sliding manifold, besides stability. Thus we only study the invariance of a close neighborhood of the sliding manifold, also to provide an estimate of the chattering due to the discrete-time discontinuous control.

Lemma 2.2.18. *Let the closed-loop system state in* (2.2.2) *be in an* $O(h^2)$*-neighborhood of the sliding manifold at* $t = t_k$*, i.e.* $\sigma_k = O(h^2)$*, but with* $\sigma_k \neq 0$*. If the discontinuous part* u^s *of the control is discretized using the explicit scheme* (2.2.5a)*, then the discretization error* ε_k *is* $O(h)$ *and the system exits the* $O(h^2)$*-neighborhood.*

Proof. Starting from equation (2.2.2) and using the control inputs $u_k^{eq} = -(CB^*)^{-1}CAx_k$ and $u_k^s = -\alpha \operatorname{sgn}(Cx_k)$, the following holds:

$$\sigma_{k+1} = C(e^{Ah} - \Psi\Pi_B A)x_k - CB^* \operatorname{sgn}(Cx_k)$$

that is $\sigma_{k+1} = \sigma_k + \Delta_k - CB^* \operatorname{sgn}(\sigma_k)$ with $\Delta_k := C(e^{Ah} - I - \Psi\Pi_B A)x_k$. Let us compute the square of the norm of the sliding variable, which is given by:

$$\|\sigma_k\|^2 + \|\Delta_k\|^2 + \|CB^* \operatorname{sgn}(\sigma_k)\|^2 + \sigma_k^T\Delta_k - \sigma_k^T CB^* \operatorname{sgn}(\sigma_k) - \Delta_k^T CB^* \operatorname{sgn}(\sigma_k). \quad (2.2.25)$$

From Lemma 2.2.17, we have $\|\Delta_k\|^2 = O(h^4)$. We can compute the order of the other terms with respect to h:

$$\|CB^*\alpha \operatorname{sgn}(\sigma_k)\|^2 = \left\|\sum_{k=0}^{\infty} h^{l+1}\frac{CA^lB}{(l+1)!}\alpha \operatorname{sgn}(\sigma_k)\right\|^2 \leq \|hCB\alpha \operatorname{sgn}(\sigma_k)\|^2 + O(h^3)$$

$$\Delta_k^T CB^*\alpha \operatorname{sgn}(\sigma_k) = O(h^3).$$

Using the Cauchy-Schwarz inequality on the remaining terms yields

$$|\sigma_k^T \Delta_k| \leq \|\sigma_k\| \|\Delta_k\|$$
$$|\sigma_k^T CB^* \operatorname{sgn}(\sigma_k)| \leq \|\sigma_k\| \|CB^* \operatorname{sgn}(\sigma_k)\|.$$

If $\|\sigma_k\| = O(h^2)$, then the above terms are $O(h^4)$ and $O(h^3)$. Hence, the dominant term in (2.2.25) is $\|hCB\alpha \operatorname{sgn}(\sigma_k)\|^2$. Let $\{\lambda_i\}$ be the spectrum of hCB, with $\lambda_m = \min_i |\lambda_i|$ and $\lambda_M = \max_i |\lambda_i|$. We have the following:

$$\lambda_m h\alpha \sqrt{p} \leq \|hCB \operatorname{sgn}(\sigma_k)\| \leq \lambda_M h\alpha \sqrt{p}.$$

Inserting this in (2.2.25) yields that $\|\sigma_{k+1}\|$ is $O(h)$.

With a matched perturbation, the discrete-time dynamics of σ are given by (2.2.17). Then $\|\sigma_{k+1}\|^2$ involves all the terms in (2.2.25), plus the following ones:

$$\begin{aligned}
|\sigma_k^T Cp_k| \leq \|\sigma_k\| \|Cp_k\| &= O(h^3) \text{ by Cauchy-Schwarz} \\
|\Delta_k^T Cp_k| \leq \|\Delta_k\| \|Cp_k\| &= O(h^3) \text{ by Cauchy-Schwarz} \\
(CB^* \alpha \operatorname{sgn}(\sigma_k))^T Cp_k &= O(h^2) \\
\|Cp_k\|^2 &= O(h^2).
\end{aligned}$$

Thus the dominant terms are $\|Cp_k\|$, $\|CB^* \alpha \operatorname{sgn}(\sigma_k)\|$ and $(CB^* \alpha \operatorname{sgn}(\sigma_k))^T Cp_k$, all of them $O(h)$. Those terms induce chattering and they all have the same order with respect to the sampling period h. □

Therefore in the nominal case, with an explicit discretization of u^s, the main error comes from the discretization of the discontinuous control u^s, since it increases the error by an order h. When there is a matched perturbation, it also introduce terms in $O(h)$. The dominant contribution is difficult to determine, *a priori*, but what remains is the (known) fact that the gain of the controller plays a role in the magnitude of the sliding variable.

IMPLICIT DISCRETIZATION In the following, we are interested in studying the discretization error in the same context as for the previous lemma.

Lemma 2.2.19. *Let the closed-loop system be in the discrete-time sliding phase. In the nominal case, if the discontinuous part u^s of the control is discretized using an implicit scheme, then the total discretization error ε_k has the same order as the discretization error $\Delta\sigma_k$ on u^{eq} (that is h^2 for the methods* (2.2.4a) *and* (2.2.4b)*, and h^3 for the midpoint method* (2.2.4c)*). If there is a matched perturbation, then the order is* 1 *and this increase of the order is due to the perturbation.*

Proof. Let $\Delta_k = C(e^{Ah} - I)x_k + C\Psi B u_k^{eq}$, with u_k^{eq} given by any method in (2.2.4a)-(2.2.4c). The system is supposed to be in the discrete-time sliding phase, that is $u_k^s \in (-\alpha, \alpha)^p$. Then $\widetilde{\sigma}_{k+1} = \sigma_k + CB^* u_k^s = 0$. From (2.2.2) one has:

$$\sigma_{k+1} = \sigma_k + \Delta_k + CB^* u_k^s = \Delta_k. \tag{2.2.26}$$

Let us go through all the possible discretization methods for u^{eq}: from Lemma 2.2.17, we know that in both the implicit and explicit cases, Δ_k is $O(h^2)$ and with the midpoint method (2.2.4c) it is $O(h^3)$. When there is a perturbation, we add Cp_k to (2.2.26):

$$\sigma_{k+1} = \sigma_k + \Delta_k - CB^* \operatorname{sgn}(\sigma_{k+1}) + Cp_k = \Delta_k + Cp_k.$$

The chattering due to the perturbation is predominant and the sliding variable is $O(h)$. □

In Proposition 2.2.11, conditions were given to ensure that the system stays in the discrete-time sliding phase once it reaches it, with or without perturbation. With the controller from Section 2.2.1 the discretization error $\Delta_k = 0$: in this case the chattering is solely due to the perturbation.

The simulation results of Section 3.4 indicate that the choice of discretization method has a clear incidence on the closed-loop behavior. The analysis of the experimental data will show that the order 1 can be recovered experimentally: this is what we observe in Figure 4.3 for Figure 4.5 from experimental results.

2.2.4 Proofs of the propositions in Section 2.2.2

Proof of Proposition 2.2.15

Let us recall Proposition 2.2.15

Proposition. *Consider the discrete-time closed-loop system given by* (2.2.17) *and* (2.2.6). *Let* $\{h_n\}_{n\in\mathbb{N}}$ *be any strictly decreasing sequence of positive numbers converging to* 0 *and with* $h_0 < h^*$ *(see Lemma 2.2.6). Suppose that the perturbation* $\xi\colon \mathbb{R} \to \mathbb{R}^p$ *is uniformly continuous, that CB is positive definite and that* $\alpha > 0$ *is chosen such that the conditions of Proposition 2.2.11 are satisfied for each sampling period* h_n. *Then for any interval* $S \subseteq [T^*, \infty)$, $\lim_{h_n \to 0} \|u^s - u^s_{\text{cont}}\|_{\infty,S} = 0$.

Proof. Let $\{t_k\}$ be a sequence such that for all $k \in \mathbb{N}$, $t_{k+1} - t_k = h_n$ with $t_0 = \inf S$. For the sake of clarity, we omit to write explicitly the dependence on n. From Proposition 2.2.11

and Lemma 2.2.6, we know that for all k such that $t_k \geq T^*$, $u_k^s = -(CB^*)^{-1}Cp_{k-1} = -(CB^*)^{-1}C\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B\xi(\tau)\mathrm{d}\tau$. During the sliding phase, the continuous-time controller satisfies $u_{\mathrm{cont}}^s(t) = -\xi(t)$. Let S be any time interval contained in $[T^*, +\infty)$. Let $t \in S$ and $k \in \mathbb{N}$ is such that $t \in [t_k, t_{k+1})$. Hence $u^s(t) = u_k^s$. Let us study $u_k^s - u_{\mathrm{cont}}^s(t)$:

$$\begin{aligned} u_k^s - u_{\mathrm{cont}}^s(t) &= -(CB^*)^{-1}C\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B\xi(\tau)\mathrm{d}\tau + \xi(t) \\ &= -(CB^*)^{-1}\left(\int_{t_{k-1}}^{t_k} Ce^{A(t_k-\tau)}B\xi(\tau)\mathrm{d}\tau - CB^*\xi(t)\right). \end{aligned}$$

Using (2.2.3), we obtain:

$$u_k^s - u_{\mathrm{cont}}^s(t) = -(CB^*)^{-1}\left(\int_{t_{k-1}}^{t_k} Ce^{A(t_k-\tau)}B\xi(\tau)\mathrm{d}\tau - \int_{t_k}^{t_{k+1}} Ce^{A(t_{k+1}-\tau)}B\xi(t)\mathrm{d}\tau\right) \quad (2.2.27)$$

With the change of variable $\tau' = \tau + h_n$ in the first integral, we can group the two integrals in (2.2.27) as follows:

$$\begin{aligned} u_k^s - u_{\mathrm{cont}}^s(t) &= -(CB^*)^{-1}\left(\int_{t_k}^{t_{k+1}} Ce^{A(t_{k+1}-\tau)}B(\xi(\tau - h_n) - \xi(t)\mathrm{d}\tau\right) \\ &= -(CB^*)^{-1}\Bigg(\int_{t_k}^{t_{k+1}} CB(\xi(\tau - h_n) - \xi(t)) \\ &\quad + \sum_{l=1}^{\infty}\frac{CA^lB}{l!}\left((t_{k+1}-\tau)^l(\xi(\tau - h_n) - \xi(t))\right)\mathrm{d}\tau\Bigg). \end{aligned} \quad (2.2.28)$$

Using again (2.2.3), the first factor can be approximated by:

$$\begin{aligned} (CB^*)^{-1} &= \left(h_nCB + \sum_{l=1}^{\infty}\frac{CA^lB}{(l+1)!}h_n^{l+1}\right)^{-1} \\ &= \left(I + \sum_{l=1}^{\infty}\frac{(CB)^{-1}CA^lB}{(l+1)!}h_n^l\right)^{-1}(h_nCB)^{-1} \\ &= \left(I - \frac{(CB)^{-1}CAB}{2}h_n + O(h_n^2)\right)(h_nCB)^{-1}. \end{aligned} \quad (2.2.29)$$

The Taylor expansion holds if $\sum_{l=1}^{\infty}\frac{(CB)^{-1}CA^lB}{(l+1)!}h_n^l$ has all its eigenvalues in the unit disk. This is a mere technical restriction, since it is always possible to find a small enough positive number h_{n_0} such this condition is satisfied. Since we are interested in the case where $\{h_n\}$ converges to 0, this requirement is supposed to hold. For the first term in the integrand in (2.2.28), we apply the mean value theorem for integration. If ξ is a vector-valued

function (this is the case when the sliding variable has dimension greater than 1), we apply the theorem for each component separately. This yields for $i = 1, \dots, n$ $(\int_{t_k}^{t_{k+1}} \xi(\tau - h_n) - \xi(t)\mathrm{d}\tau)_i = h_n(\xi_i(t_i' - h_n) - \xi_i(t))$ for some $t_i' \in [t_k, t_{k+1}]$. For the second part of the integrand in (2.2.28), we exchange the summation and integral signs. This is possible since the matrix exponential is normally convergent and ξ is bounded on any interval $[t_k, t_{k+1}]$ (ξ is continuous). Moreover for all $l \geq 1$, with $\tau \in [t_k, t_{k+1}]$, $(t_{k+1} - \tau)^l(\xi(\tau - h_n) - \xi(t)) = O(h_n)$. Thus $\int_{t_k}^{t_{k+1}} (t_{k+1} - \tau)^l(\xi(\tau - h_n) - \xi(t))\mathrm{d}\tau = O(h_n^2)$. Then (2.2.28) can be rewritten as:

$$u_k^s - u_{\mathrm{cont}}^s(t) = -(I + O(h_n))\Big(\int_{t_k}^{t_{k+1}} h_n^{-1}(\xi(\tau - h_n) - \xi(t))\mathrm{d}\tau + O(h_n)\Big).$$

Taking the supremum norm and using standard estimation yields:

$$\begin{aligned}\|u_k^s - u_{\mathrm{cont}}^s(t)\|_\infty &\leq \|I + O(h_n)\|_\infty \left(\max_i \sup_{t' \in [t_k, t_{k+1}]} |\xi_i(t' - h) - \xi_i(t)| + \|O(h_n)\|_\infty\right) \\ &\leq \max_i \sup_{t' \in [t_k, t_{k+1}]} |\xi_i(t' - h_n) - \xi_i(t)| + O(h_n). \qquad (2.2.30)\end{aligned}$$

Since ξ is uniformly continuous, for every $\varepsilon > 0$, there exists $\delta > 0$ such that for all $t_1, t_2 \in \mathbb{R}$, $|t_1 - t_2| \leq \delta$ implies $\|\xi(t_1) - \xi(t_2)\| \leq \varepsilon$. Then the right-hand side of (2.2.30) converges to 0 as $h_n \to 0$. Since this is true for all $t \in S$, the proof is complete. □

Proof of Proposition 2.2.16

Proposition. *Suppose that CB is positive definite, and ξ is a real-valued continuously differentiable with bounded derivative function. Let $\{h_n\}_{n \in \mathbb{N}}$ be any strictly decreasing sequence of positive numbers converging to 0 with $h_0 < h^*$. Let α be chosen such that the conditions of Proposition 2.2.11 are verified for each h_n. Let $T > T^*$ with T^* defined in Proposition 2.2.11. Then $\lim_{h_n \to 0} \mathrm{Var}_{T^*}^T(u^s) = \mathrm{Var}_{T^*}^T(u_{\mathrm{cont}}^s)$.*

Proof. Let $\{t_k\}$ be a sequence such that for all $k \in \mathbb{N}$, $t_{k+1} - t_k = h_n$ with $T^* + h_n > t_0 \geq T^*$ and N the largest integer such that $t_N \leq T$. Let us recall that, with the implicit controller defined in Equations (2.2.17) and (2.2.6), the reaching phase duration is bounded and that the sliding phase is established at $t = T^*$ if h_n is small enough. Recall that in continuous time, $u_{\mathrm{cont}}^s \equiv -\xi$ in the sliding phase. Let us study the difference between the variations of u^s and u_{cont}^s:

$$\mathrm{Var}_{T^*}^T(u^s) - \mathrm{Var}_{T^*}^T(u_{\mathrm{cont}}^s) = \sum_k \|u_{k+1}^s - u_k^s\| - \int_{T^*}^T \|\dot{\xi}(\tau)\|\mathrm{d}\tau,$$

Without loss of generality, the integral is taken in the Riemannian sense. Hence for all $\varepsilon > 0$, it is possible to find an index n_0 such that for all $n \geq n_0$,

$$\int_{T^*}^{T} \|\dot{\xi}(\tau)\| \mathrm{d}\tau = h_n \sum_k \|\dot{\xi}(t_k)\| + \int_{T^*}^{t_0} \|\dot{\xi}(\tau)\| \mathrm{d}\tau + \int_{t_N}^{T} \|\dot{\xi}(\tau)\| \mathrm{d}\tau \pm \varepsilon.$$

This leads to the estimation

$$\left| \mathrm{Var}_{T^*}^{T}(u^s) - \mathrm{Var}_{T^*}^{T}(u^s_{\mathrm{cont}}) \right| \leq \sum_k \left| \|u^s_{k+1} - u^s_k\| - h_n \|\dot{\xi}(t_k)\| \right| + \varepsilon + 2h_n \mathcal{M},$$

with $\mathcal{M} > 0$ a majorant of $\|\dot{\xi}(\tau)\|$ on $[T^*, T]$. Using the reverse triangle inequality yields

$$\leq \sum_k \|u^s_{k+1} - u^s_k - h_n \dot{\xi}(t_k)\| + \varepsilon + 2h_n \mathcal{M}.$$

Expanding the difference $u^s_{k+1} - u^s_k$, we get

$$\begin{aligned} u^s_{k+1} - u^s_k = (CB^*)^{-1} \Bigg(& \int_{t_k}^{t_{k+1}} CB(\xi(\tau) - \xi(\tau - h_n)) \mathrm{d}\tau \\ & + \int_{t_k}^{t_{k+1}} C(e^{A(t_{k+1}-\tau)} - I) B(\xi(\tau) - \xi(\tau - h_n)) \mathrm{d}\tau \Bigg). \end{aligned}$$

Using the approximation for $(CB^*)^{-1}$ in (2.2.29) yields

$$\begin{aligned} u^s_{k+1} - u^s_k - h_n \dot{\xi}(t_k) = (I + O(h_n)) \Bigg(& \int_{t_k}^{t_{k+1}} \frac{\xi(\tau) - \xi(\tau - h_n)}{h_n} - \dot{\xi}(t_k) \mathrm{d}\tau \\ & + \int_{t_k}^{t_{k+1}} (CB)^{-1} C(e^{A(t_{k+1}-\tau)} - I) B \frac{\xi(\tau) - \xi(\tau - h_n)}{h_n} \mathrm{d}\tau \Bigg). \end{aligned} \tag{2.2.31}$$

The first integral can be transformed into

$$\int_{t_k}^{t_{k+1}} h_n^{-1}(\xi(\tau) - \xi(\tau - h_n)) - \dot{\xi}(\tau) \mathrm{d}\tau + \int_{t_k}^{t_{k+1}} \dot{\xi}(\tau) - \dot{\xi}(t_k) \mathrm{d}\tau. \tag{2.2.32}$$

Using the (uniform) continuity of $\dot{\xi}$ on the bounded interval $[T^*, T]$, the second integral in (2.2.32) can be bounded by

$$\left\| \int_{t_k}^{t_{k+1}} \dot{\xi}(\tau) - \dot{\xi}(t_k) \mathrm{d}\tau \right\| \leq h_n \varepsilon',$$

for any $\varepsilon' > 0$ independent of the interval $[t_k, t_{k+1}]$, by increasing the starting index of the sequence $\{h_n\}$ if necessary. Using standard estimation inequalities, we obtain the following estimate for the second integral in (2.2.31):

$$\begin{aligned} & \left\| \int_{t_k}^{t_{k+1}} C(e^{A(t_{k+1}-\tau)} - I) B \frac{\xi(\tau) - \xi(\tau - h_n)}{h_n} \mathrm{d}\tau \right\| \\ & \quad \leq \int_{t_k}^{t_{k+1}} \|C\| \|B\| (e^{\|A\| h_n} - 1) \left\| \frac{\xi(\tau) - \xi(\tau - h_n)}{h_n} \right\| \mathrm{d}\tau. \end{aligned}$$

Let us now come back to our main computation

$$\begin{aligned}\left|\mathrm{Var}_{T^*}^{T}(u^{\varsigma}) - \mathrm{Var}_{T^*}^{T}(u_{\mathrm{cont}}^{\varsigma})\right| &\leq \varepsilon + 2h_n\mathcal{M} \\ &+ (1 + O(h_n))\Big(\int_{t_0}^{t_N} \frac{\|\xi(\tau) - \xi(\tau - h_n) - h_n\dot{\xi}(\tau)\|}{h_n}\mathrm{d}\tau \\ + \sum_k h_n\varepsilon' + \|(CB)^{-1}\|\|C\|\|B\|(e^{\|A\|h_n} - 1)&\int_{t_0}^{t_N} \frac{\|\xi(\tau) - \xi(\tau - h_n)\|}{h_n}\mathrm{d}\tau\Big).\end{aligned} \tag{2.2.33}$$

The continuity of ξ, $\dot{\xi}$ and $\|\cdot\|$ allows us to use the mean value theorem for integration on both integrals:

$$\begin{aligned}\int_{t_0}^{t_N} \frac{\|\xi(\tau) - \xi(\tau - h_n) - h_n\dot{\xi}(\tau)\|}{h_n}\mathrm{d}\tau = \\ (t_N - t_0)\frac{\|\xi(\tau') - \xi(\tau' - h_n) - h_n\dot{\xi}(\tau')\|}{h_n}.\end{aligned}$$

Using Taylor's theorem with the remainder in its Lagrange form on ξ, we can write

$$\xi(\tau' - h_n) = \xi(\tau') - h_n\dot{\xi}(\tau') + h_n^2\dot{\xi}(\tau_t),$$

with $\tau_t \in [\tau' - h_n, \tau']$. Hence, this integral is $O(h_n)$. Switching to the second one, we have

$$\begin{aligned}\int_{t_0}^{t_N} \frac{\|\xi(\tau) - \xi(\tau - h_n)\|}{h_n}\mathrm{d}\tau = \\ (t_N - t_0)\frac{\|\xi(\tau'') - \xi(\tau'' - h_n)\|}{h_n} \xrightarrow[h_n\to 0]{} \|\dot{\xi}(\tau''')\| \leq \mathcal{M}\end{aligned}$$

for some τ', τ'', $\tau''' \in [t_0, t_N]$. Since $e^{\|A\|h_n} - 1 = O(h_n)$, this part of (2.2.33) vanishes as $h \to 0$, which only leaves the sum

$$\sum_k h_n\varepsilon' = \left\lfloor \frac{(T - T^*)}{h_n} \right\rfloor h_n\varepsilon' \leq (T - T^*)\varepsilon'.$$

Hence we can rewrite (2.2.33) as

$$\left|\mathrm{Var}_{T^*}^{T}(u^{\varsigma}) - \mathrm{Var}_{T^*}^{T}(u_{\mathrm{cont}}^{\varsigma})\right| \leq \varepsilon + (T - T^*)\varepsilon' + O(h_n),$$

where ε and ε' can be set arbitrarily small by cutting the sequence $\{h_n\}$. This completes the proof. □

2.3 Twisting Controller

Let us study the second type of "square fully discontinuous" controller: the twisting algorithm that we introduced in Section 1.1.2. This section is organized as follows: we first provide a discrete-time version of the algorithm based on an implicit discretization. The matrix of the AVI we obtain is not a P-matrix: the analysis carried out in the previous section cannot be applied directly. Hence, we derive some results on F-uniqueness of AVI, a concept introduced in Definition 2.1.11. This enables us to further analyze the implicitly discretized twisting algorithm. While doing so, we find some shortcomings of this controller: the implicit discretization does not yield a system free of numerical chattering. Therefore, we slightly modify the structure of the control law to prevent this issue, which enables us to show that on the double integrator system, this controller ensures the global finite-time Lyapunov stability of the origin.

2.3.1 Discrete-time twisting controller

We now study the discrete-time version of the twisting controller, whose continuous-time dynamics is

$$\begin{aligned} \ddot{\sigma} &= a(x,t) + g_s(x,t)u \\ -u &\in a\,\mathrm{Sgn}(\sigma) + b\,\mathrm{Sgn}(\dot{\sigma}), \end{aligned}$$

with $a > b > 0$. Let consider the case of a double integrator, that is

$$\ddot{\sigma} \in -a\,\mathrm{Sgn}(\sigma) - b\,\mathrm{Sgn}(\dot{\sigma}).$$

Recasting this as a first order system, we get

$$\dot{\Sigma} = A\Sigma + B\lambda \quad \text{with} \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ a & b \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix} \tag{2.3.1}$$

$$\text{and} \quad \Sigma \coloneqq \begin{pmatrix} \sigma \\ \dot{\sigma} \end{pmatrix}, -\lambda \in \mathrm{Sgn}\begin{pmatrix} \sigma \\ \dot{\sigma} \end{pmatrix} = \mathrm{Sgn}\,\Sigma.$$

Let us discretize the dynamics using the ZOH method. The discontinuous control input is implemented with the semi-implicit method as:

$$\Sigma_{k+1} = A^*\Sigma_k + B^*\lambda_{k+1} \quad \text{with} \quad \lambda_{k+1} = \begin{pmatrix} \lambda_{1,k+1} \\ \lambda_{2,k+1} \end{pmatrix} \tag{2.3.2}$$

$$-\lambda_{1,k+1} \in \mathrm{Sgn}(\sigma_{k+1}),\ -\lambda_{2,k+1} \in \mathrm{Sgn}(\dot{\sigma}_{k+1}) \tag{2.3.3}$$

$$\text{and} \quad \mathcal{A}^* = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}, B^* = h \begin{pmatrix} \frac{h}{2}a & \frac{h}{2}b \\ a & b \end{pmatrix} = h \begin{pmatrix} \frac{h}{2} \\ 1 \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix}.$$

The discrete-time dynamics for each sliding variable are given by

$$\begin{cases} \sigma_{k+1} = \sigma_k + h\dot{\sigma}_k + \dfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) \\ \dot{\sigma}_{k+1} = \dot{\sigma}_k + h(a\lambda_{1,k+1} + b\lambda_{2,k+1}). \end{cases}$$

Like with the classical SMC case, we use the equivalence

$$-\lambda_{k+1} \in \operatorname{Sgn}(\Sigma_{k+1}) \quad \Longleftrightarrow \quad -\Sigma_{k+1} \in N_H(\lambda) \quad \text{with } H \coloneqq [-1,1]^2, \tag{2.3.5}$$

given in Fact A.1.18, to rewrite (2.3.2) and (2.3.3) as the generalized equation

$$0 \in \mathcal{A}\Sigma_k + B^*\lambda + N_H(\lambda). \tag{2.3.6}$$

The sliding variables and control input are then given by:

$$\Sigma_{k+1} = \mathcal{A}\Sigma_k + B^*\lambda \quad \text{and} \quad u_k = a\lambda_1 + b\lambda_2. \tag{2.3.7}$$

Let us now study the existence and uniqueness of a solution to this AVI.

Lemma 2.3.1. *The AVI* (2.3.6) *has always a solution.*

Proof. Since the mapping $z \mapsto B^*z + \mathcal{A}\Sigma_k$ is continuous and H is compact, we can apply Theorem 2.1.9. □

The study of uniqueness is not as easy as with the classical SMC. This is due to the fact that B^* is not a P-matrix and not even positive-semidefinite. To see this, let us compute the symmetric part of B^*:

$$2B_s^* = B^* + B^{*T} = h \begin{pmatrix} ha & a + \frac{h}{2}b \\ a + \frac{h}{2}b & 2b \end{pmatrix},$$

whose determinant is equal to

$$h\left(2abh - \left(a + \tfrac{h}{2}b\right)^2\right) = -h\left(a - \tfrac{h}{2}b\right)^2 < 0.$$

Thus the matrix B^* is indefinite. We could try to reformulate the AVI (2.3.6) into an LCP and see if the w-uniqueness (remember Theorem 2.1.6) property holds, but this does not work either. In the following, we derive F-uniqueness results for some classes of AVI that are an extension of those for LCP, in the sense that the condition on the matrix is the same or close to the one listed in Theorem 2.1.6.

2.3.2 F-uniqueness of AVI

Let us recall the condition on the matrix $\mathcal{M}$ of the LCP$(q, \mathcal{M})$ for the w-uniqueness to hold:

Definition 2.3.2. [27, p. 155] Given a matrix $\mathcal{M} \in \mathbb{R}^{n\times n}$, we say that any vector whose sign is reversed by $\mathcal{M}$ belongs to the nullspace of $\mathcal{M}$ if the following inequality holds

$$[z_i(\mathcal{M}z)_i \leq 0 \text{ for all } i \in \{1, \dots, n\}] \quad \Longrightarrow \quad [\mathcal{M}z = 0].$$

The results that we present here depend on the set and the matrix associated with the AVI, but not the constant term q. Let us start by considering the case where the set is $\mathcal{B}_\infty$, the unit ball for the maximum norm before moving on to more generic sets.

Proposition 2.3.3. *Consider an* AVI$(\mathcal{B}_\infty, q, \mathcal{M})$ *in* $\mathbb{R}^n$ *with* $\mathcal{B}_\infty$ *the unit ball for the maximum norm. If every vector whose sign is reversed by* $\mathcal{M}$ *belongs to the nullspace of* $\mathcal{M}$*, then the AVI enjoys the F-uniqueness property.*

Proof. We rewrite the AVI$(\mathcal{B}_\infty, q, \mathcal{M})$ as the generalized equation $0 \in \mathcal{M}z+q+N_{\mathcal{B}_\infty}(z)$. Let $w := \mathcal{M}z + q$ and recall from Fact A.1.17 that

$$-w \in N_{\mathcal{B}_\infty}(z) \quad \Longleftrightarrow \quad -z \in \operatorname{Sgn}(w).$$

Note that $\operatorname{Sgn} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a maximal monotone operator with the remarkable property that it is component-wise maximal monotone:

$$\text{for all } i \in \{1, \dots, n\} \quad \operatorname{Sgn}_i(w) = \operatorname{Sgn}(w_i).$$

This observation is pivotal to the proof. Now suppose that the AVI has multiple solutions and let z^1, z^2 be any two of them with $\Delta z := z^1 - z^2$. Let $\Delta w := w^1 - w^2 = \mathcal{M}\Delta z$. Using the definition of the maximal monotonicity gives:

$$\langle z_i^1 - z_i^2, w_i^1 - w_i^2\rangle \leq 0 \qquad \text{for all } i \in \{1, \dots, n\}$$

or more compactly written

$$(\Delta z)_i(\mathcal{M}\Delta z)_i \leq 0 \qquad \text{for all } i \in \{1, \dots, n\}.$$

This implies that the sign of Δz is reversed by $\mathcal{M}$. Hence by the hypothesis on $\mathcal{M}$, Δz belongs to its nullspace and therefore $\Delta w = 0$, completing the proof. □

Let us now extend this to a bigger class of AVI by considering convex compact polytopes other than $\mathcal{B}_\infty$. To do so, we need to recast the equivalence like (2.3.5) into a more general framework. Fact A.1.17 states this relation in the more general setting of a relation between the subdifferentials of an indicator function $\delta(\cdot)$ and a support function $\mathfrak{h}(\cdot)$. Therefore we shall now state the equivalence (2.3.5) as

$$-w \in N_{\mathcal{B}_\infty}(z) = \partial\delta_{\mathcal{B}_\infty}(z) \iff z \in \partial\mathfrak{h}_{\mathcal{B}_\infty}(-w).$$

The next proposition deals with AVI where the set is polytopic but anymore $\mathcal{B}_\infty$ or even box-shaped.

Proposition 2.3.4 (F-uniqueness of AVI). *Consider an* AVI(K, q, M) *in* $\mathbb{R}^n$ *with* K *a non-empty convex polytope. If there exists a nondegenerate linear transformation, with a matrix representation* $L \in \mathbb{R}^{n\times n}$*, from* $\mathcal{B}_\infty$ *to* K*, and if every vector whose sign is reversed by* $\widetilde{M} := L^T ML$ *belongs to the nullspace of* $\widetilde{M}$*, then the AVI enjoys the F-uniqueness property.*

Proof. We try to show that the statement holds by transforming the AVI(K, q, M) into an AVI$(\mathcal{B}_\infty, \widetilde{q}, \widetilde{M})$ and applying the previous result. The existence of the linear transformation $L \in \mathbb{R}^{n\times n}$ enables us to write that for every $z \in K$, there exists a unique $y \in \mathcal{B}_\infty$ such that $z = Ly$. The core part in the transformation between the two AVIs is how to relate the normal cones of the set K and $\mathcal{B}_\infty$. Let us start by stating the relation between the indicator functions of the two sets: for all pair z and y such that $z = Ly$,

$$\delta_K(z) = \delta_{\mathcal{B}_\infty}(y) = \delta_{\mathcal{B}_\infty}(L^{-1}z),$$

since L is nondegenerate. Using the chain rule (Theorem A.1.15), we get

$$\begin{aligned} \partial\delta_K(z) &= L^{-T}\partial\delta_{\mathcal{B}_\infty}(L^{-1}z) \\ L^T\partial\delta_K(z) &= \partial\delta_{\mathcal{B}_\infty}(L^{-1}z) \end{aligned}$$

Rewriting this using normal cones, we have

$$L^T N_K(z) = N_{\mathcal{B}_\infty}(L^{-1}z). \tag{2.3.8}$$

Now we transform the AVI(K, q, M)

$$0 \in Mz + q + N_K(z). \tag{2.3.9}$$

by noticing that z is solution to the previous AVI if and only if it is a solution to

$$0 \in L^T M z + L^T q + L^T N_K(z). \tag{2.3.10}$$

Using (2.3.8) and moving to the y variable, we get

$$0 \in L^T M L y + L^T q + N_{\mathcal{B}_\infty}(y).$$

Letting $\widetilde{M} := L^T M L$ and $\widetilde{q} := L^T q$, we obtain

$$0 \in \widetilde{M} y + \widetilde{q} + N_{\mathcal{B}_\infty}(y).$$

We can apply Proposition 2.3.3 on this AVI given the hypothesis on $\widetilde{M}$. Hence, we get that for any solution y of this AVI, $\widetilde{M} y + \widetilde{q}$ is unique, and if we have two distinct solutions y^1 and y^2, the difference $\Delta y := y^1 - y^2$ is in $\ker \widetilde{M}$. This amount to the F-uniqueness of the AVI (2.3.10) and the fact that there exists two distinct solutions z^1 and z^2, the difference $\Delta z := z^1 - z^2$ is in $\ker L^T M = \ker M$ since L is nonsingular. Therefore $Mz + q$ is unique for all solutions z of the AVI (2.3.9) and the proof is complete. □

Remark 2.3.5. It does not seem easy to characterize the matrices L and M such that every vector whose sign is reversed by $L^T M L$ belongs to the nullspace of $L^T M L$. Let us consider the case when M enjoys this property and is a rank-one matrix, that is $M = vu^T$ for some $u, v \in \mathbb{R}^n$. Now the matrix $L^T M L = L^T v u^T L = (Lv)(Lu)^T = \tilde{v}\tilde{u}^T$ is another rank-one matrix. By construction, the columns of $L^T M L$ are linearly dependent, therefore this matrix is P_0 if for all i, $\tilde{v}_i \tilde{u}_i \geq 0$. The case when a diagonal element of $L^T M L$ is 0 requires special attention: the condition on the determinant implies that the column must be the zero vector. This prevents any element of $\tilde{u}$ from being zero.

Remark 2.3.6. Reasoning along the same lines, if we impose that M (or $L^T M L$) is a P-matrix, then we get the uniqueness of the AVI(K, q, M). This result is already available in the first case, when the set K is box-shaped. To the best of our knowledge, the second proposition, when K is a polytope has not been studied before.

Remark 2.3.7. In Section 2.1.2, we recall a result given for VI(K, F) regarding the F-uniqueness: the function $F(\cdot)$ has to be pseudo-monotone on K, that is for all x and y in K,

$$(x - y)^T F(y) \geq 0 \quad \Rightarrow \quad (x - y)^T F(x) \geq 0. \tag{2.3.11}$$

This condition is not necessary fulfilled with an indefinite matrix that ensures the F-uniqueness of the AVI. Let us illustrate this fact with an AVI over the square $[-1, 1]^2$

and with the following rank-one matrix

$$M = \begin{pmatrix} 2 & 3 \\ 4 & 6 \end{pmatrix} \quad \text{whose symmetric part is:} \quad M_s \coloneqq \begin{pmatrix} 2 & 7/2 \\ 7/2 & 6 \end{pmatrix}.$$

The matrix M is indefinite since $\operatorname{tr} M_s = 8$ and $\det M_s = -1/4$. The condition (2.3.11) fails with $y = 0$ and x the eigenvector associated with the negative eigenvalue. In the same Section 2.1.2, we present what is, to the best of our knowledge, the most general result for F-uniqueness for AVI(K, q, M), which requires the matrix M to be psd-plus. Hence the positive semi-definiteness of M is a necessary condition for the results from the literature to apply. Note that the F-uniqueness of this AVI can be established using Proposition 2.3.3.

2.3.3 Analysis of the implicit twisting controller

Let us now continue with the analysis of the controller, now that we can characterize the solutions to the AVI (2.3.6). We use a generic form of this AVI$([-1, 1]^2, q, M)$ where the matrix M is rank-one and given by

$$M = c \begin{pmatrix} \frac{b}{2} \\ 1 \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix}, \tag{2.3.12}$$

with $c > 0$ a given constant and the gain $a > b > 0$ and the sampling period $h > 0$. No constraints are given on the vector q. This formulation allows for dynamics other than the simple double integrator. This was the case while testing the twisting controller on the electropneumatic experimental setup of Section 4.1. Let us recall that Σ_{k+1} is defined as $q + M\lambda_{k+1}$, with λ_{k+1} a solution to the AVI.

Lemma 2.3.8. *Suppose that the dynamics of a system with state Σ_k and a control input given by the implicit twisting controller, can be recast as the* AVI$(\mathcal{B}_\infty, q, M)$ *with M as in* (2.3.12)*. Then for a given vector q, both Σ_{k+1} and the control input value u_k are unique. Moreover if $\Sigma_{k+1} \neq 0$, then the solution of the AVI is also unique.*

Proof. The uniqueness of Σ_{k+1} is a direct application of Proposition 2.3.3: by construction M is rank-one, and with 2 positive diagonal elements. Then by Definition 2.1.5 it is P_0. Its two columns are linearly dependent: the matrix M fulfills the requirements of Proposition 2.3.3.

For the uniqueness of the control input value, remember from Proposition 2.3.3 that if the AVI(K, q, M) has more than one solution, then the difference $\Delta\lambda$ between two

solutions has to lie in $\ker \mathcal{M}$, that is span $(b, -a)^T$ in our case. Note that in (2.3.7), u_k is defined as $(a, b)\lambda$. It is easy to see that any element of $\ker \mathcal{M}$ is orthogonal to $(a, b)^T$, which ensures uniqueness of u_k.

For the last part of the statement, if λ^1 and λ^2 are solutions to the AVI, then they are both in $\partial \mathfrak{h}_{[-1,1]^2}(-\Sigma_{k+1})$, by the inverse relation given in Fact A.1.17 and the uniqueness of Σ_{k+1}. Furthermore, we know from Fact A.1.14 that λ^1 and λ^2 are such that

$$\langle \lambda^1, -\Sigma_{k+1} \rangle = \langle \lambda^2, -\Sigma_{k+1} \rangle = \mathfrak{h}_{[-1,1]^2}(-\Sigma_{k+1}).$$

Whence $\Delta\lambda := \lambda^1 - \lambda^2$ is orthogonal to Σ_{k+1}. Remember that

$$\Delta\lambda \in \ker \mathcal{M} = \operatorname{span} \begin{pmatrix} -b \\ a \end{pmatrix},$$

which means that $\Sigma_{k+1} = s\begin{pmatrix} a \\ b \end{pmatrix}$, with $s \in \mathbb{R}$. We are interested in the case $\Sigma_{k+1} \neq 0$ which imposes that $s \neq 0$. Given that $a > b > 0$, Σ_{k+1} belongs to either the first or the third quadrant. Hence,

$$\partial \mathfrak{h}_{[-1,1]^2}(-\Sigma_{k+1}) = \arg \sup_{y \in [-1,1]^2} \langle y, -\Sigma_{k+1} \rangle = \begin{cases} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} & \text{if } s < 0 \\ \left\{ \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\} & \text{if } s > 0. \end{cases}$$

The two sets in the right-hand are both singletons: the solution λ of the AVI is therefore unique whenever $\Sigma_{k+1} \neq 0$. □

Remark 2.3.9. The same results hold in the continuous-time twisting algorithm (1.1.6), where the selections $-\lambda_1 \in \operatorname{Sgn}(\sigma)$ and $-\lambda_2 \in \operatorname{Sgn}(\dot{\sigma})$ are uniquely defined, except when $u = 0$. In this case the values lie on the segment defined by $a\lambda_1 + b\lambda_2 = 0$ and $\lambda_1 \in [-1, 1]$. It is also noteworthy that the set described by the last equality is span $(b - a)^T \cap [-1, 1]^2 = \ker \mathcal{M} \cap [-1, 1]^2$. This is related to the fact that B as in (2.3.1) and $\mathcal{M}$ have the same nullspace.

Let us now go back to the study of the double integrator with the twisting algorithm. For the reader's convenience its dynamics is given again here:

$$\begin{cases} \sigma_{k+1} = \sigma_k + h\dot{\sigma}_k + \dfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) & \text{(2.3.13a)} \\ \dot{\sigma}_{k+1} = \dot{\sigma}_k + h(a\lambda_{1,k+1} + b\lambda_{2,k+1}). & \text{(2.3.13b)} \end{cases}$$

and in matrix form, we have

$$\Sigma_{k+1} = A^*\Sigma_k + B^*\lambda_{k+1} \quad \text{with} \quad A^* = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}, B^* = h\begin{pmatrix} \frac{h}{2}a & \frac{h}{2}b \\ a & b \end{pmatrix} = h\begin{pmatrix} \frac{h}{2} \\ 1 \end{pmatrix}\begin{pmatrix} a & b \end{pmatrix}. \tag{2.3.14}$$

With the twisting algorithm, there are two switching surfaces

$$S_1 = \{(\sigma, \dot{\sigma}) \in \mathbb{R}^2 \mid \sigma = 0\}$$
$$S_2 = \{(\sigma, \dot{\sigma}) \in \mathbb{R}^2 \mid \dot{\sigma} = 0\}.$$

In continuous-time, we know that in each quadrant, the vector field is directed onto one of them, but neither S_1 nor S_2 are invariant, if the conditions on the gains a and b are fulfilled. In discrete-time, one natural question is whether this non-invariance property is preserved. We unfortunately have to answer by the negative.

Lemma 2.3.10. *The segment S_c of S_1 defined by*

$$\left\{(0, \dot{\sigma}) \colon |\dot{\sigma}| \leq \frac{h}{2}(a - b)\right\}$$

is invariant for the system with dynamics (2.3.13) *coupled with the implicitly discretized twisting controller.*

Proof. Let us initialize the system in S_c, that is $\sigma_k = 0$ and $|\dot{\sigma}_k| \leq \frac{h}{2}(a - b)$. We proceed as follows: we show that there exists $\lambda_{k+1} \in [-1, 1]^2$ such that $\dot{\sigma}_{k+1} = -\dot{\sigma}_k$ and $-\lambda_{k+1} \in \mathrm{Sgn}(\Sigma_{k+1})$. The previous proposition ensures the uniqueness of Σ_{k+1}, u_k and λ_{k+1} which saves us from checking for other control input values.

Let us compute the value of λ_{k+1} such that $\sigma_{k+1} = 0$: using the discrete-time dynamics (2.3.13), we get

$$0 = h\dot{\sigma}_k + \tfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+2}),$$

and the control input value is

$$u_k = a\lambda_{1,k+1} + b\lambda_{2,k+1} = -\tfrac{2}{h}\dot{\sigma}_k. \tag{2.3.15}$$

Reporting this in (2.3.13b), we obtain

$$\dot{\sigma}_{k+1} = -\dot{\sigma}_k. \tag{2.3.16}$$

Let us check that there exists $-\lambda_{k+1} \in \mathrm{Sgn}(\Sigma_{k+1})$ which gives the control input value in (2.3.15). Injecting the condition on $\dot{\sigma}_k$ in (2.3.15), we have:

$$-a + b \leq u_k \leq a - b. \tag{2.3.17}$$

The value of $\lambda_{2,k+2}$ can be anything in $[-1, 1]$. However since $\sigma_{k+1} = 0$, $\lambda_{1,k+1}$ can take any value in $[-1, 1]$. Since $a > b > 0$, it is always possible to pick $\lambda_{1,k+1}$ such that $a\lambda_{1,k+1} + b\lambda_{2,k+2}$ takes any value in the range given by (2.3.17). Then we have a control input that steers the system from $(0, \dot{\sigma}_k)$ to $(0, -\dot{\sigma}_k)$ if $|\dot{\sigma}_k| \leq \frac{h}{2}(a - b)$. □

The relation (2.3.16) indicates that cycling can occur for the discrete-time twisting we study. Recalling that the B^* matrix in (2.3.14) has rank one, we know that its range has dimension one. In the following we study the set of points that can reach the origin, which is the control objective.

Lemma 2.3.11. *The origin of system* (2.3.13) *is only reachable from the line segment*

$$S_0 := \left\{(\sigma_k, \dot{\sigma}_k) \in \mathbb{R}^2 \colon \sigma_k + \frac{h}{2}\dot{\sigma}_k = 0, |\dot{\sigma}_k| \leq h(a+b)\right\}. \tag{2.3.18}$$

Proof. Let us first study the set of points Σ_k such that $\Sigma_{k+1} = (0, 0)$. Using the recurrence relations (2.3.13a) and (2.3.13b), we get the system

$$\begin{cases} \sigma_k + h\dot{\sigma}_k + \dfrac{h}{2}(ah\lambda_{1,k+1} + bh\lambda_{2,k+1}) = 0 & \text{(2.3.19a)} \\ \dot{\sigma}_k + ah\lambda_{1,k+1} + bh\lambda_{2,k+1} = 0. & \text{(2.3.19b)} \end{cases}$$

Inserting (2.3.19b) in (2.3.19a) yields $\sigma_k = \dfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1})$. Combining with (2.3.19b) we get

$$\sigma_k + \frac{h}{2}\dot{\sigma}_k = 0. \tag{2.3.20}$$

Hence the origin can only be reached from the hyperplane defined by (2.3.20). Since λ_{k+1} is in $[-1, 1]^2$, the relation (2.3.19b) gives the additional constraints $|\dot{\sigma}_k| \leq h(a+b)$. □

Now that we studied how the origin of the closed-loop system can be reached, let us investigate the set of initial conditions which can reach the origin.

Proposition 2.3.12. *Consider system* (2.3.13). *The set of initial positions that can reach the origin is a union of countably many segments. Therefore it has measure zero.*

Proof. Lemma 2.3.11 characterizes the pre-image of the origin as the line segment S_0. To further study the set of points that can reach the origin, we compute the successive pre-images of this segment. We now show that each pre-image is a union of segments. We shall proceed by induction with the statement been that the set of points reaching the origin in N steps is a union of segments, which have a finite negative slope. The segment S_0 satisfies those requirements. Now we tackle the inductive step of the proof. We want to compute the pre-image is a union of segments. First we consider the intersections of a segment with the interior of each quadrant. We consider only the non-empty ones, which

are themselves segments. The control input u_k is then constant for each point of the preimage. Let us suppose that the intersection admits the following representation: $\dot{\sigma}_{k+1} = c_k\sigma_{k+1} + d_k$. Therefore we are interesting in solving the following linear system:

$$\begin{cases} \sigma_{k+1} = \sigma_k + h\dot{\sigma}_k + \dfrac{h^2}{2}u_k & \text{(2.3.21a)} \\ c_k\sigma_{k+1} + d_k = \dot{\sigma}_k + hu_k. & \text{(2.3.21b)} \end{cases}$$

Since $c_k < 0$ by the statement, we have $1 - hc_k \neq 0$ and computing (2.3.21a) $- h$(2.3.21b), we get

$$\sigma_{k+1} = \frac{1}{1-hc_k}\sigma_k + \frac{hd_k}{1-hc_k} + \frac{h^2}{2(1-hc_k)}u_k. \tag{2.3.22}$$

Using (2.3.21b), one obtains

$$\dot{\sigma}_k = c_k\sigma_{k+1} + d_k - hu_k.$$

Substituting σ_{k+1} with its value in (2.3.22) gives

$$\dot{\sigma}_k = \underbrace{\frac{c_k}{1-hc_k}}_{c_{k+1}}\sigma_k + \underbrace{\frac{d_k}{1-hc_k} - h\frac{2+hc_k}{2(1-hc_k)}}_{d_{k+1}}u_k. \tag{2.3.23}$$

Let us check that if $c_k < 0$, then $c_k < c_{k+1} < 0$. From (2.3.23)

$$\frac{1}{c_{k+1}} = \frac{1}{c_k} - h,$$

Therefore, c_k is always finite, negative, monotonically increasing and upper bounded by 0. Hence the preimage of this intersection is defined by the linear transformation (2.3.23) and is therefore a segment, with a negative slope. Now we have to deal with the points that are on the surface S_1 or S_2. Let us compute the preimage of such a point. We start with the case $\Sigma_{k+1} \in S_1$: $\sigma_{k+1} = 0$, $\dot{\sigma}_{k+1} \neq 0$. The system of equations is

$$\begin{cases} 0 = \sigma_k + h\dot{\sigma}_k + \frac{h^2}{2}u_k & \text{(2.3.24a)} \\ \dot{\sigma}_{k+1} = \dot{\sigma}_k + hu_k. & \text{(2.3.24b)} \end{cases}$$

We use the value of the control input in (2.3.24b) in (2.3.24a) to get:

$$\begin{aligned} 0 &= \sigma_k + h\dot{\sigma}_k + \tfrac{h}{2}\dot{\sigma}_{k+1} - \tfrac{h}{2}\dot{\sigma}_k \\ \tfrac{h}{2}\dot{\sigma}_k &= -\sigma_k - \tfrac{h}{2}\dot{\sigma}_{k+1} \\ \dot{\sigma}_k &= -\tfrac{2}{h}\sigma_k - \dot{\sigma}_{k+1} \end{aligned}$$

The bound on u_k define a segment on this line of negative slope, which has a point in S_1 as image. If $\Sigma_{k+1} \in S_2$, we have $\sigma_{k+1} \neq 0$, $\dot{\sigma}_{k+1} = 0$.

$$\begin{cases} \sigma_{k+1} = \sigma_k + h\dot{\sigma}_k + \frac{h^2}{2} u_k & \text{(2.3.25a)} \\ 0 \quad = \dot{\sigma}_k + h u_k. & \text{(2.3.25b)} \end{cases}$$

As before, we use the value of the control input in (2.3.25b) in (2.3.25a) to get:

$$\begin{aligned} \sigma_{k+1} &= \sigma_k + \tfrac{h}{2}\dot{\sigma}_k \\ \dot{\sigma}_k &= -\tfrac{2}{h}\sigma_k - \tfrac{2}{h}\sigma_{k+1} \end{aligned}$$

Again the segment that has for image a point in S_2 has a finite negative slope.

Thus, every point in the set that reach the origin in $N+1$ steps belongs to a segment of negative slope. Then all the pre-images of the origin are defined as an union of sets of measure zero, union of line segments. Invoking the countably subadditivity of a measure completes the proof. □

Remark 2.3.13. From (2.3.16) in the proof of Lemma 2.3.10, one can see that if the system does not start from very specific initial conditions, at some points it may cycle between two values $(0, \alpha)$ and $(0, -\alpha)$ with $|\alpha| < (ah + bh)/2$, but does not reach the origin. It is not possible to control the value of α: it varies with both the initial condition and the sampling period. Here are some numerical experiments that illustrate this effect.

On Figure 2.1, we can see some simulation results illustrating this phenomenon. On the left picture, it is easy to see that the system can end up cycling far from the origin. On the right picture, the segment S_0, defined in (2.3.18), in yellow and going through the origin, can be seen as well as some of its preimages. The logarithmic scale better illustrates that the initial conditions reaching the origin are of measure zero as stated in Proposition 2.3.12. The cycling behavior induces a numerical chattering, which is surely undesirable. In the next section, we present a modified discrete-time twisting algorithm for which the origin is globally finite-time reachable and which has the same computational complexity as the implicitly discretized algorithm.

2.3.4 Modified implicit twisting controller

The main issue with the implicit discretization of the twisting algorithm is that the segment S_0 is reachable only from initial conditions with measure zero in $\mathbb{R}^2$. We want that the

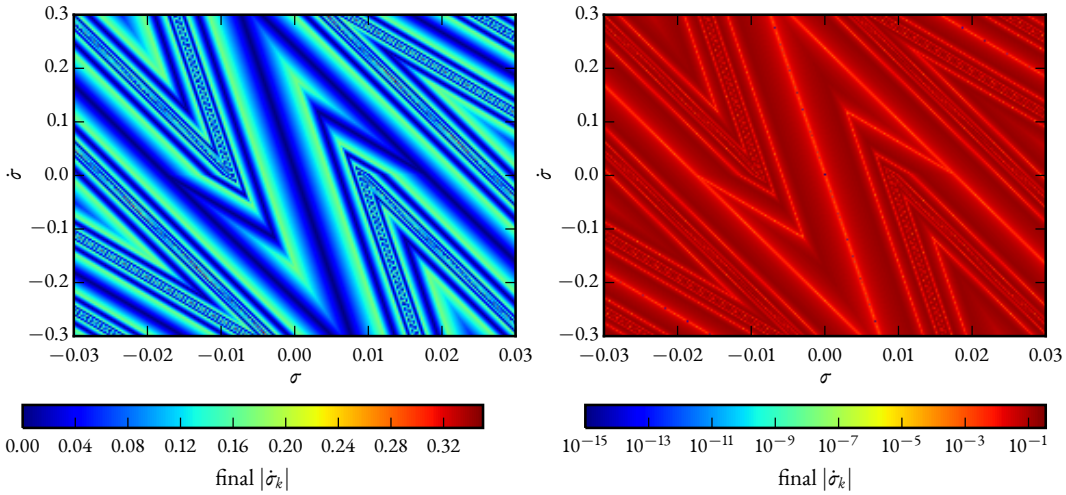


Figure 2.1: Asymptotic behavior of the double integrator with the implicit twisting. One million simulations with initial conditions in $[-0.03, 0.03] \times [-0.3, 0.3]$ were run. Each initial position is then colored with the value of $|\dot{\sigma}_k|$ at the end. The state cycles in all cases but one, due to the finite numerical precision: the state magnitude in this case is close to the machine precision. The sampling period is 50ms $a = 10$ and $b = 3$. The only difference between the two pictures is the scale: linear on the left and logarithmic on the right.

trajectories starting from any initial conditions reach the origin. Therefore, we present a modified twisting algorithm, featuring a different set for its AVI, which is globally finite-time stable.

The basic idea is to consider the λ variable to be defined as

$$-\lambda_{k+1} \in \partial \mathfrak{h}_{-K}(\Sigma_{k+1}), \tag{2.3.26}$$

with K a bounded polytopic convex set. This allows us to extend the inclusion

$$-\lambda_{k+1} \in \mathrm{Sgn}(\Sigma_{k+1}),$$

which given that $\mathrm{Sgn}(\cdot) = \partial \mathfrak{h}_{[-1,1]^2}(\cdot)$ can be rewritten as

$$-\lambda_{k+1} \in \partial \mathfrak{h}_{[-1,1]^2}(\Sigma_{k+1}).$$

Using this approach based on AVI is interesting since we want to be able to design a control law that steers a set with positive measure to S_0. To achieve this, we can modify the set K, where λ takes value, such that, at least, an half-line containing a part of $-S_0$ is one of the normal cone to K. We consider the case where K is a convex polytope, defined as

$$K = \{x \in \mathbb{R}^2 \mid Ex \leq b\} \quad \text{with } E \in \mathbb{R}^{4\times 2} \text{ and } b \in \mathbb{R}^4. \tag{2.3.27}$$

From Fact A.1.10, we know that the normal cone is generated by the rows of E. Hence, a simple way to have an half-line containing a part of $-S_0$ as at least one of the normal cones, is to include as a constraint, that is at least one row of E has to be proportional to $(h/2, -1)$. To be more concrete, the square $[-1, 1]^2$ admits the representation

$$\left\{x \in \mathbb{R}^2 \mid Hx \leq b\right\} \quad \text{with } H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

We propose to use the matrix

$$E = \begin{pmatrix} 1 & 0 \\ -h/2 & 1 \\ -1 & 0 \\ h/2 & -1 \end{pmatrix}$$

in (2.3.27). The choice of the vector b depends on the constraints we want to impose on the control inputs. Let us discuss three possible choices:

$$b_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 1 \\ 1 - h/2 \\ 1 \\ 1 - h/2 \end{pmatrix} \quad b_3 = \begin{pmatrix} 1 \\ 1 + h/2 \\ 1 \\ 1 + h/2 \end{pmatrix}.$$

With b_1, we obtain a parallelogram, which is not contained in the square $[-1, 1]^2$. If the original control constraints were important to respect, then by using the vector b_2, the resulting set is a parallelogram contained in the unit square. Finally another choice could be b_3, which gives a set containing the original square. Note that all those sets are symmetric with respect to the origin.

In the following, we still consider that the sliding variables have the dynamics (2.3.13) but now the discontinuous control variables are defined by the inclusion (2.3.26) with the set K defined as (2.3.27) with the choice b_1. The transformation from $[-1, 1]^2$ to K is given by

$$L = \begin{pmatrix} 1 & 0 \\ \frac{b}{2} & 1 \end{pmatrix}.$$

The control input is still given as

$$u_k = a\lambda_{1,k+1} + b\lambda_{2,k+1}. \tag{2.3.28}$$

The wellposedness of the controller is now investigated.

Lemma 2.3.14. *The system composed of the double integrator system and modified implicit twisting controller as defined in* (2.3.26) *enjoys the uniqueness of* Σ_{k+1} *and* u_k*. Moreover, if* $\Sigma_{k+1} \neq 0$*, then* λ_{k+1} *is also unique.*

Proof. The dynamics of the system is the same as in Lemma 2.3.8. Hence the uniqueness of Σ_{k+1} and u_k follows from the same type of arguments as the proof of this lemma. We just need to show that the conditions for the F-uniqueness of the AVI are fulfilled. As noted in Remark 2.3.5, since B^* as in (2.3.14) is a rank-one matrix, $L^T B^* L$ is the rank-one matrix given by

$$L^T B^* L = h \begin{pmatrix} 1 & \frac{h}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{h}{2} \\ 1 \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{h}{2} & 1 \end{pmatrix} = h \begin{pmatrix} h \\ 1 \end{pmatrix} \begin{pmatrix} a + \frac{h}{2} b & b \end{pmatrix}.$$

This matrix is singular with positive diagonal elements. By construction its columns are linearly dependent. Hence, we have by Proposition 2.3.4 the uniqueness of Σ_{k+1}. As shown in the proof of this proposition, this implies that the difference between any two solutions is in $\ker \mathcal{M}$. As with the implicitly discretized twisting controller, this ensures the uniqueness of the control input value u_k. For the last part of the statement, the uniqueness of Σ_{k+1} implies that if λ^1 and λ^2 are two distinct solutions, then their opposite are both in the set $\partial \mathfrak{h}_{-K}(\Sigma_{k+1})$. Furthermore, we know from Fact A.1.14 that this means that

$$\langle -\lambda^1, \Sigma_{k+1} \rangle = \langle -\lambda^2, \Sigma_{k+1} \rangle = \mathfrak{h}_{-K}(\Sigma_{k+1}).$$

Whence $\Delta\lambda := \lambda^1 - \lambda^2$ is orthogonal to Σ_{k+1}. But we know that

$$\Delta\lambda \in \ker \mathcal{M} = \operatorname{span} \begin{pmatrix} -b \\ a \end{pmatrix},$$

which means that $\Sigma_{k+1} = s \begin{pmatrix} a \\ b \end{pmatrix}$, with $s \in \mathbb{R}$. We are interested in the case $\Sigma_{k+1} \neq 0$ which means that $s \neq 0$. Given that $a > b > 0$, we get that:

$$\partial \mathfrak{h}_{-K}(\Sigma_{k+1}) = \arg \sup_{y \in -K} \langle y, \Sigma_{k+1} \rangle = \begin{cases} \left\{ \begin{pmatrix} 1 \\ 1 + \frac{h}{2} \end{pmatrix} \right\} & \text{if } s > 0 \\ \left\{ \begin{pmatrix} -1 \\ -1 - \frac{h}{2} \end{pmatrix} \right\} & \text{if } s < 0. \end{cases}$$

The two sets in the right-hand are both singletons: the solution λ of the AVI is therefore unique whenever $\Sigma_{k+1} \neq 0$. □

Let us now turn our attention to the stability analysis. The conditions on the gain now read:

$$a > \left(1 + \frac{h}{2}\right) b > 0,$$

which comes from the forthcoming Lyapunov analysis. Before searching for a candidate Lyapunov function, let us provide some relations between the variables used in the modified twisting controller (2.3.13) and (2.3.26). First note that we can link the support functions of the set K and $[-1,1]^2$:

$$\mathfrak{h}_{-K}(x) = \sup_{y \in -K} \langle y, x \rangle = \sup_{z \in -[-1,1]^2} \langle Lz, x \rangle = \sup_{z \in [-1,1]^2} \langle z, L^T x \rangle = \mathfrak{h}_{[-1,1]^2}(L^T x).$$

Using the chain rule (Theorem A.1.15), we get: for all $\Sigma \in \mathbb{R}^2$,

$$\partial \mathfrak{h}_{-K}(\Sigma) = L \partial \mathfrak{h}_{[-1,1]^2}(L^T \Sigma).$$

Thus the relation (2.3.26) can be rewritten equivalently as:

$$-\lambda_{1,k} \in \operatorname{Sgn}(\sigma_k + \frac{h}{2}\dot{\sigma}_k) \tag{2.3.29}$$

$$-\lambda_{2,k} \in \operatorname{Sgn}(\dot{\sigma}_k) + \frac{h}{2} \operatorname{Sgn}(\sigma_k + \frac{h}{2}\dot{\sigma}_k) \tag{2.3.30}$$

This gives rise to the following bounds:

$$|\lambda_{1,k+1}| \leq 1 \tag{2.3.31}$$

$$|\lambda_{2,k+1}| \leq 1 + \tfrac{h}{2} \tag{2.3.32}$$

$$|u_k| \leq a + b(1 + \tfrac{h}{2}). \tag{2.3.33}$$

Now let us compute the set of points reaching the origin in one step.

Lemma 2.3.15. *The origin of the double integrator with the controller defined as* (2.3.26) *is only reachable from the line segment*

$$S_0' := \left\{(\sigma_k, \dot{\sigma}_k) \in \mathbb{R}^2 \colon \sigma_k + \tfrac{h}{2}\dot{\sigma}_k = 0, |\dot{\sigma}_k| \leq h(a + b(1 + \tfrac{h}{2}))\right\}.$$

Proof. With the same computation as in the proof of Lemma 2.3.11, the origin can only be reached from the hyperplane defined by $\sigma_k + \frac{h}{2}\dot{\sigma}_k = 0$. Since λ_{k+1} is in K, the new upper bound on u_k given in (2.3.33) gives the constraint $|\dot{\sigma}_k| \leq h(a + b(1 + \frac{h}{2}))$. □

Now let us provide additional relations between the controller and sliding variables. The inclusion (2.3.29) can be inverted as:

$$\sigma_k + \frac{h}{2}\dot{\sigma}_k \in N_{[-1,1]}(-\lambda_{1,k})$$

$$\text{or} \quad \text{for all } \lambda_1' \in [-1,1], \quad (\lambda_1' + \lambda_{1,k})\left(\sigma_k + \frac{h}{2}\dot{\sigma}_k\right) \leq 0. \tag{2.3.34}$$

Note that from (2.3.29), we have

$$\left|\sigma_k + \frac{h}{2}\dot{\sigma}_k\right| = -\lambda_{1,k}\left(\sigma_k + \frac{h}{2}\dot{\sigma}_k\right).$$

Also if $h < 2$, the relation (2.3.30) implies that

$$-\lambda_{2,k}\dot{\sigma}_k > 0 \text{ if } \dot{\sigma}_k \neq 0 \quad \text{and} \quad -\lambda_{2,k}\dot{\sigma}_k = 0 \text{ if } \dot{\sigma}_k = 0, \tag{2.3.35}$$

since for $\dot{\sigma}_k \neq 0$, the first term in the right-hand side of (2.3.30) always dominates the second one.

Now that we have those relations ready for use, let us look for a candidate Lyapunov function. We shall try the following function

$$V_k = V(\sigma_k, \dot{\sigma}_k) = a\left|\sigma_k + \frac{h}{2}\dot{\sigma}_k\right| + \frac{1}{2}\dot{\sigma}_k^2 - \frac{h}{2}b\lambda_{2,k}\dot{\sigma}_k. \tag{2.3.36}$$

Introducing the variable λ_{k+1} yields

$$\begin{aligned} &= -a\lambda_{1,k}(\sigma_k + \frac{h}{2}\dot{\sigma}_k) + \frac{1}{2}\dot{\sigma}_k^2 - \frac{h}{2}b\lambda_{2,k}\dot{\sigma}_k \\ &= \begin{pmatrix} -a\lambda_{1,k} & \frac{1}{2}(\dot{\sigma}_k - ah\lambda_{1,k} - hb\lambda_{2,k}) \end{pmatrix} \Sigma_k. \end{aligned} \tag{2.3.37}$$

Starting from (2.3.36) and using (2.3.35), it is easy to assess that if $h < 2$, then $V(\cdot)$ is positive everywhere except at the origin where it vanishes, and that it is also radially unbounded. The remaining part is to prove that this function decreases between two iterates, that is $\Delta V_k \coloneqq V_{k+1} - V_k < 0$. Let us first recall the dynamics of the system:

$$\begin{cases} \sigma_{k+1} = \sigma_k + h\dot{\sigma}_k + \frac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) & \text{(2.3.38a)} \\ \dot{\sigma}_{k+1} = \dot{\sigma}_k + h(a\lambda_{1,k+1} + b\lambda_{2,k+1}). & \text{(2.3.38b)} \end{cases}$$

First note that using (2.3.38b) in (2.3.37), we can write

$$V_{k+1} = -a\lambda_{1,k+1}\sigma_{k+1} + \tfrac{1}{2}\dot{\sigma}_k\dot{\sigma}_{k+1}.$$

Now we investigate the evolution of V:

$$\Delta V_k = -a\lambda_{1,k+1}\left(\sigma_k + h\dot{\sigma}_k + \tfrac{h^2}{2}\overbrace{(a\lambda_{1,k+1} + b\lambda_{2,k+1})}^{\star}\right) + a\lambda_{1,k}\sigma_k$$

$$+ \tfrac{1}{2}(\dot\sigma_k\dot\sigma_{k+1} - \dot\sigma_k(\underbrace{\dot\sigma_k}_{\star} - ah\lambda_{1,k} - hb\lambda_{2,k})).$$

Using (2.3.38b), we substitute the terms tagged with $\star$ to get

$$\Delta V_k = a(\lambda_{1,k} - \lambda_{1,k+1})\sigma_k - \tfrac{ah}{2}\lambda_{1,k+1}(\dot\sigma_{k+1} + \dot\sigma_k) + \tfrac{h}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1} + a\lambda_{1,k} + b\lambda_{2,k})\dot\sigma_k$$

$$= a(\lambda_{1,k} - \lambda_{1,k+1})(\sigma_k + \tfrac{h}{2}\dot\sigma_k) + \tfrac{h}{2}(-a\lambda_{1,k+1}\dot\sigma_{k+1} + (a\lambda_{1,k+1} + b\lambda_{2,k+1} + b\lambda_{2,k})\dot\sigma_k)$$

We replace $\dot\sigma_{k+1}$ with its expression in (2.3.38b).

$$= a(\lambda_{1,k} - \lambda_{1,k+1})(\sigma_k + \tfrac{h}{2}\dot\sigma_k) + \tfrac{h}{2}(-ah\lambda_{1,k+1}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) + (b\lambda_{2,k+1} + b\lambda_{2,k})\dot\sigma_k).$$

Using again relation (2.3.38b) to replace the term $\lambda_{2,k+1}\dot\sigma_k$, we get

$$\Delta V_k = a(\lambda_{1,k} - \lambda_{1,k+1})(\sigma_k + \tfrac{h}{2}\dot\sigma_k) + \tfrac{h}{2}\Big(-ah\lambda_{1,k+1}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) + b\lambda_{2,k+1}\dot\sigma_{k+1}$$

$$- hb\lambda_{2,k+1}(a\lambda_{1,k+1} + b\lambda_{2,k+1})) + b\lambda_{2,k}\dot\sigma_k\Big).$$

A final rearrangement in the second term yields

$$\Delta V_k = a(\lambda_{1,k} - \lambda_{1,k+1})(\sigma_k + \tfrac{h}{2}\dot\sigma_k) - \tfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1})^2 + \tfrac{bh}{2}(\lambda_{2,k+1}\dot\sigma_{k+1} + \lambda_{2,k}\dot\sigma_k). \tag{2.3.39}$$

Let us analyze the last equality term by term: using the fact that $\sigma_k + \dfrac{h}{2}\dot\sigma_k \in N_{[-1,1]}(-\lambda_{1,k})$ and using (2.3.34), the first term is nonpositive. The second term is clearly nonpositive and the third one too, using the relation (2.3.35). To conclude the proof, let us show that the variation ΔV_k is negative as long as the origin is not reached. The second term in (2.3.39) is zero only if

$$a\lambda_{1,k+1} + b\lambda_{2,k+1} = 0. \tag{2.3.40}$$

Using (2.3.31) and (2.3.32) we get that with the condition $a > (1+h/2)b$, the latter equation has a solution if and only if $|\lambda_{1,k+1}| < 1$. Thus we have (2.3.40) if and only if

$$\sigma_{k+1} + \frac{h}{2}\dot\sigma_{k+1} = 0. \tag{2.3.41}$$

Using (2.3.40) in the dynamics (2.3.38b), we get that

$$\dot\sigma_{k+1} = \dot\sigma_k.$$

Going back to the analysis of (2.3.39), using (2.3.35), the last term in the right-hand side is zero if and only if

$$\dot{\sigma}_{k+1} = \dot{\sigma}_k = 0,$$

which combined with (2.3.41) implies that

$$\sigma_{k+1} = 0,$$

and by (2.3.29) that $\sigma_k = 0$. Whence ΔV_k can be zero only when the system has already reached the origin.

For the finite-time stability, remember that the three terms in the right-hand side of (2.3.39) are all nonpositive. Hence we can find an upper bound of ΔV_k by considering only one of those terms. Let us take a closer look at the second one:

$$-\tfrac{h^2}{2}(a\lambda_{1,k+1} + b\lambda_{2,k+1}) = \tfrac{h^2}{2}u_k.$$

Note that since $a > b > 0$, as long as $\lambda_{1,k+1} \neq 0$, which is true for $\sigma_{k+1} + \frac{h}{2}\dot{\sigma}_{k+1}$, it holds that $|u_k| \leq a - b(1 - \frac{h}{2})$. Hence, if this condition on the sliding variable holds, then we have

$$\Delta V_k \leq -\frac{h^2}{2}(a - b(1 - \tfrac{h}{2}))^2. \tag{2.3.42}$$

From Lemma 2.3.15, we know that if the state of the system belongs to S_0', the origin is reached at the next time instant t_{k+2}. So if we prove that this segment is reachable in finite-time from any initial conditions, then the origin is globally finite-time reachable. Hence, given that (2.3.42) hold everywhere except on the line (2.3.41), just need to bound ΔV_k away from 0 if σ_{k+1} belongs to the line (2.3.41) minus S_0'. For this reason, suppose that Σ_{k+1} belongs to the line (2.3.41) and that $|\dot{\sigma}_{k+1}| > h(a + b)$. In the third term we have the following expression

$$\lambda_{2,k+1}\dot{\sigma}_{k+1} = -|\dot{\sigma}_{k+1}| - \frac{h}{2}\lambda_{1,k+1}\dot{\sigma}_{k+1} \leq -(1 - \frac{h}{2})|\dot{\sigma}_{k+1}| \leq -h(1 - \frac{h}{2})(a + b).$$

Hence we have the estimate

$$\Delta V_k \leq -\frac{bh^2}{2}(1 - \frac{h}{2})(a + b). \tag{2.3.43}$$

For all k, ΔV_k is smaller than the maximum of the right-hand side of (2.3.42) and (2.3.43). Both quantities being negative constants, the finite-time convergence to S_0' holds, hence the finite-time convergence to the origin. We summarize the developments in the following proposition.

Proposition 2.3.16. *Let $h < 2$. The origin is the unique equilibrium of the system* (2.3.38) *with the controller given by* (2.3.26) *and is globally Lyapunov finite-time stable.*

Remark 2.3.17. This discrete-time Lyapunov function V is close to the one used in [88]: $a|\sigma| + \dot{\sigma}^2/2$.

Corollary 2.3.18. *Let $h < 2$. The origin is the unique equilibrium of the continuous-time system* (2.3.1) *with the controller given by* (2.3.26) *and is globally Lyapunov finite-time stable.*

Proof. The ZOH discretization being exact, we know by the previous proposition that there exists $k_0 \in \mathbb{N}$ such that $\Sigma_{k_0} = 0$. Then for all $k > k_0$, $\Sigma_k = 0$, with the control input $u_k = 0$, as we can easily infer from (2.3.38). On each sampling interval, the continuous-time system has the dynamics

$$\dot{\Sigma} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \Sigma + \begin{pmatrix} 0 & 0 \\ a & b \end{pmatrix} 0 = 0.$$

This concludes the proof. □

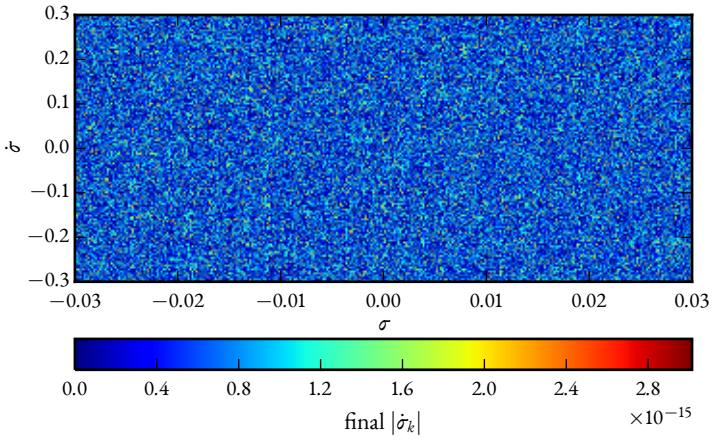


Figure 2.2: Asymptotic behavior of the double integrator with the modified implicit twisting. The setup is the same as in Figure 2.1: one million simulations with initial conditions in $[-0.03, 0.03] \times [-0.3, 0.3]$ were run. Each initial position is then colored with the value of $|\dot{\sigma}_k|$ at the end. The sampling period is 50ms $a = 10$ and $b = 3$.

Let us finish this study of the modified discrete-time twisting controller by showing simulation results mirroring the ones given in Figure 2.1 for the implicitly discretized twisting controller. On Figure 2.2, it is clear that for any initial condition, the system is

steered to the origin. Note that the scaling factor for the final values of $\dot{\sigma}_k$ is 10^{-15}. The values of $\dot{\sigma}_k$ are close to the machine precision and no pattern can be seen. The cycling problem has been sucessfully removed and the stability result given by Proposition 2.3.16 is also illustrated.

2.4 Sliding Mode Observers

The problem of observation being closely related to the control problem, we touch upon the sliding mode observer topic. We first recall the continuous-time version, before discretizing the observer using the same spirit. Then we switch to a version based on the equivalence-based control. Then we tackle the choice of the matrix gain before illustrating with simulation results.

Let us consider a linear time-invariant system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B\xi(t,x) \\ y(t) = Cx(t), \end{cases} \tag{2.4.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$. We further assume that B is full column rank and that C is full row rank. The system is supposed to be observable.

The time instants at which the measurements are made form the sequence $\{t_k\}$, $k \in \mathbb{N}$. The available measurements are $y_k := Cx_k = Cx(t_k)$. We suppose that the control function u is piecewise constant and that we have the knowledge of the time where it changes, such that we can compute the quantity

$$B_k^* := \int_{t_k}^{t_{k+1}} \exp(A\tau)Bu(\tau)\mathrm{d}\tau$$

for all time instants t_k of interest.

2.4.1 Utkin's observer

As in [33], we apply the state transformation $T := \begin{bmatrix} C \\ H_C \end{bmatrix}$, where H_C spans the null space of C. The matrix T is invertible by the full row-rank assumption. This defines

$$TAT^{-1} =: \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \bar{A}$$

and

$$TB =: \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \bar{B}.$$

Let $z \in \mathbb{R}^{n-p}$ and $Tx := [y, z]^T$. The variable z is the unknown part of the state x. In the new coordinate system, C and H_C are transformed into $\bar{C} := \begin{bmatrix} I_p & 0_{p\times n-p} \end{bmatrix}$ and $\bar{N}_C := \begin{bmatrix} 0_{n-p\times p} & I_{n-p} \end{bmatrix}$. We can rewrite the dynamical part of the system (2.4.1) as

$$\dot{y}(t) = A_{11}y(t) + A_{12}z(t) + B_1u(t) \tag{2.4.2}$$

$$\dot{z}(t) = A_{21}y(t) + A_{22}z(t) + B_2u(t). \tag{2.4.3}$$

The sliding mode observer proposed by Utkin [107] has the structure

$$\begin{cases} \dot{\hat{y}}(t) = A_{11}\hat{y}(t) + A_{12}\hat{z}(t) + B_1u(t) - \nu(t) \\ \dot{\hat{z}}(t) = A_{21}\hat{y}(t) + A_{22}\hat{z}(t) + B_2u(t) - \bar{L}\nu(t) \\ \nu(t) \in \mathcal{M}\,\mathrm{Sgn}(\hat{y}(t) - y(t)), \end{cases} \tag{2.4.4}$$

with $\mathcal{M} \in \mathbb{R}_+$ and $\bar{L} \in \mathbb{R}^{n-p\times p}$ the gain matrix. The design of the observer consists of choosing the matrix $\bar{L}$ such that the estimate $\hat{z} \to z$.

2.4.2 Discrete-time version of the observer

We present now the discretized version of the observer, using Zero-Order Hold (ZOH) and an implicit discretization of the $\mathrm{Sgn}(\cdot)$ multifunction in (2.4.4). Rewriting the equations (2.4.2) and (2.4.3) in discrete time, with a ZOH discretization, yields

$$y_{k+1} = \bar{C}\bar{A}^* \begin{bmatrix} y_k \\ z_k \end{bmatrix} + \bar{C}\bar{B}_k^* \tag{2.4.5}$$

$$z_{k+1} = \bar{N}_C\bar{A}^* \begin{bmatrix} y_k \\ z_k \end{bmatrix} + \bar{N}_C\bar{B}_k^*, \tag{2.4.6}$$

where $\bar{A}^* = \exp(\bar{A}h)$ and $\bar{B}_k^* := TB_k^*$. To simplify the discrete-time dynamics, let us slightly modify the observer dynamics by introducing the term Ψ^{-1} before the nonsmooth part of the dynamics, with $\Psi := \int_0^h \exp(\bar{A}\tau)\mathrm{d}\tau$. The rational for this modification will become clear in the forthcoming analysis. Then the observer state $\hat{x} := \begin{bmatrix} \hat{y} & \hat{z} \end{bmatrix}^T$ has the continuous-time dynamics

$$\dot{\hat{x}}(t) = \bar{A}\hat{x}(t) + \bar{B}u(t) - \Psi^{-1} \begin{bmatrix} I_p \\ \bar{L} \end{bmatrix} \nu(t), \tag{2.4.7}$$

with $\nu\colon \mathbb{R} \to \mathbb{R}^p$ a piecewise constant function, changing at time t_k. By integration over $[t_k, t_{k+1})$, we obtain the recurrence relations:

$$\hat{x}_{k+1} = \bar{A}^* \hat{x}_k + B_k^* - \begin{bmatrix} I_p \\ \bar{L} \end{bmatrix} \nu_k \tag{2.4.8}$$

and:

$$\hat{y}_{k+1} = \bar{C}\bar{A}^* \begin{bmatrix} \hat{y}_k \\ \hat{z}_k \end{bmatrix} + \bar{C}B_k^* - \nu_k \tag{2.4.9}$$

$$\hat{z}_{k+1} = \bar{N}_C \bar{A}^* \begin{bmatrix} \hat{y}_k \\ \hat{z}_k \end{bmatrix} + \bar{N}_C B_k^* - \bar{L}\nu_k. \tag{2.4.10}$$

We proceed in two steps. First we extract information from the measurement y at time t_k. Then we use it in the observer. For all $k \in \mathbb{N}^*$, at time t_k, we have the knowledge of y_k, y_{k-1} and $\hat{z}_k$.

Let $\widetilde{y} = \hat{y} - y$ and $\widetilde{z} = \hat{z} - z$ be respectively the error on the measured (respectively reconstructed) part of the state. In the following we detail the compute of the variable ν_k.

2.4.3 Analysis of the discrete-time observer

The first approach to compute ν_k is to use a discretized version of (2.4.4). As with the SMC, the specificity of our approach is to use an implicit Euler method, sometimes called backward Euler, for the argument of the $\mathrm{Sgn}(\cdot)$ multifunction. The situation is as follows: the current time is t_{k+1} and the measurement y_{k+1} is available. Our objective is to use it to update the states of the observer, that is to determine the value of $\hat{y}_{k+1}$ and $\hat{z}_{k+1}$. The implicit discretization of ν yields

$$\nu_{k+1} = \mathcal{M}\,\mathrm{Sgn}(\hat{y}_{k+1} - y_{k+1}),$$

which added to the dynamics (2.4.9) and (2.4.10) gives the system

$$\begin{cases} \hat{x}_{k+1} = \bar{A}^* \hat{x}_k + B_k^* - \begin{bmatrix} I_p \\ \bar{L} \end{bmatrix} \nu_{k+1} \\ \nu_k \in \mathcal{M}\,\mathrm{Sgn}(\hat{y}_{k+1} - y_{k+1}). \end{cases} \tag{2.4.11}$$

In order to deterthe SMC, tine the value of ν_{k+1}, we look at the error dynamics of $\widetilde{y} = \hat{y}_{+1} - y_{k+1}$. The dynamics in (2.4.11) onto the subspace spanned by $\bar{C}$ are given by the relation

$$\begin{cases} \hat{y}_{k+1} - y_{k+1} = \bar{C}\bar{A}^* \hat{x}_k + \bar{C}B_k^* - y_{k+1} - \nu_{k+1} \\ \nu_{k+1} \in \mathcal{M}\,\mathrm{Sgn}(\hat{y}_{k+1} - y_{k+1}). \end{cases} \tag{2.4.12}$$

This system has 2 unknowns: $\hat{y}_{k+1}$ and ν_{k+1}. It has the same form has the closed-loop reduced dynamics studied in Section 2.2. The same approach applies and therefore the system (2.4.12) is non-anticipative and has a unique solution, that can be simply computed as

$$\nu_{k+1} = \mathrm{proj}_{[-\mathcal{M},\mathcal{M}]^p}(\bar{C}\bar{A}^*\hat{x}_k + \bar{C}B_k^* - y_{k+1}). \tag{2.4.13}$$

This result is due to the particular structure of the system (2.4.12), namely that the non-smooth part due to the $\mathrm{Sgn}(\cdot)$ is only left-multiplied by the positive scalar $\mathcal{M}$ in the recurrence relation (2.4.12). This is the consequence of the introduction of the term Ψ in the continuous-time dynamics (2.4.7).

Beyond solvability property, the system (2.4.12) is also Lyapunov finite-time stable, that is if $\hat{y}_0 - y_0 = \widetilde{y}_0 \neq 0$, then there exists $k_0 \in \mathbb{N}$ such that $\widetilde{y}_{k_0} = 0$. However, in the context of measurements of a linear subspace of a state, it is not clear that this situation is of interest. Indeed at the time instant t_0, start of the estimation process, one can always update the initial estimate such that $\bar{C}\hat{x}_0 = 0$. A simple procedure is to solve by least square the problem $\bar{C}\hat{x}^c = \bar{C}\hat{x}_0 - y_0$ and then to update the estimated state by $\hat{x}_0^+ = \hat{x}_0 - \hat{x}^c$. After this operation, $\bar{C}\hat{x}_0^+ = y_0$. The solution of the least square problem is given by $\hat{x}^c = \bar{C}(\bar{C}^T\bar{C})^{-1}\bar{C}^T(\bar{C}\hat{x}_0 - y_0)$, which implies that $\bar{N}_C\hat{x}^c = 0$. Therefore the value of $\widetilde{z}$ is not affected by this update.

Furthermore, the convergence of the observer is conditional on the error been small with respect to $\mathcal{M}$, see [33, p. 128]. The error $\begin{bmatrix}\widetilde{y} & \widetilde{z}\end{bmatrix}^T$ may leave the sliding manifold $\widetilde{y} = 0$ as well. This motivates us construct an observer which evolves unconditionally on this sliding manifold.

2.4.4 Equivalent-based observer

The main idea behind sliding mode observer is to constrain the error to evolve in the subspace $\widetilde{y} = 0$. The idea is to consider ν_k as a Lagrange multiplier in (2.4.9), which ensures that the constraint $\hat{y}_{k+1} = y_{k+1}$ is satisfied. Imposing this relation in (2.4.8) yields the value of ν_k:

$$\nu_k = \bar{C}\bar{A}^*\hat{x}_k + \bar{C}\bar{B}_k^* - y_{k+1}. \tag{2.4.14}$$

The right-hand side contains only known quantities at time t_{k+1}. At this point it should be noted that the value of ν_{k+1} in (2.4.13) is the one obtained from (2.4.14) and projected onto an $\mathcal{M}\mathcal{B}_\infty$. As we shall see in Section 2.4.5, ν_{k+1} contains important information on the error on the non measured part z. Therefore we will tend to use the formula in (2.4.14), since the projection onto an hypercube can be seen has a reduction of the available information the

observer gets from the measurement. We can also make a connection with the notion of equivalent part of the control used in classical sliding mode control. The latter is designed such that it makes the sliding manifold positively invariant, by forcing the derivative to be in the tangent space.

2.4.5 Pole placement

If we combine the relations in (2.4.5) and (2.4.9), we have

$$\bar{C}\bar{A}^* \begin{bmatrix} y_k \\ z_k \end{bmatrix} = \bar{C}\bar{A}^* \begin{bmatrix} \hat{y}_k \\ \hat{z}_k \end{bmatrix} - \nu_{k+1}.$$

Therefore

$$\nu_{k+1} = \bar{C}\bar{A}^* \begin{bmatrix} 0 \\ \hat{z}_k - z_k \end{bmatrix} = \bar{A}^*_{12}\tilde{z}_k, \tag{2.4.15}$$

where $\bar{A}^*$ is split as $\begin{bmatrix} \bar{A}^*_{11} & \bar{A}^*_{12} \\ \bar{A}^*_{21} & \bar{A}^*_{22} \end{bmatrix}$, with blocks of the same dimensions as for $\bar{A}$. By computing ν_{k+1} using (2.4.14), some information about the error on the non measurable state variable z is available. We use this quantity in the equation (2.4.10) to make the estimate converge to the true value.

In a similar fashion to the continuous-time case, substituting (2.4.15) in (2.4.10), and computing the error z at time t_{k+1}, by using (2.4.6) and (2.4.10), yields

$$\begin{aligned} \tilde{z}_{k+1} = \hat{z}_{k+1} - z_{k+1} &= \bar{N}_C\bar{A}^* \begin{bmatrix} 0 \\ \hat{z}_k - z_k \end{bmatrix} - \bar{L}\bar{A}^*_{12}\tilde{z}_k \\ &= (\bar{A}^*_{22} - \bar{L}\bar{A}^*_{12})\tilde{z}_k \end{aligned}$$

Using [102, Prop. 6.2.11, p. 275], if $(\bar{A}, \bar{C})$ is observable and $h(\lambda - \mu) \neq 2l\pi i$ for all $l \in \mathbb{Z}^*$, with λ, μ any two eigenvalues of A, then $(\bar{A}^*, \bar{C})$ is also observable. We can then use a result in [33, p. 128], to claim that if $(\bar{A}, \bar{C})$ is observable, then the pair $(\bar{A}^*_{22}, \bar{A}^*_{12})$ shares this property. Thus it is possible to find L to place the eigenvalues of the matrix $\bar{A}^*_{22} - L\bar{A}^*_{12}$ at any desired values. If those eigenvalues are in the unit circle, then $\tilde{z}$ shrinks exponentially fast to the origin, ensuring the convergence of the state estimate to the real one.

Finally, combining (2.4.10) and (2.4.14), the update equations for the observer are

$$\hat{y}_{k+1} = y_{k+1}$$

$$\begin{aligned}\hat{z}_{k+1} &= \bar{A}^*_{21}\hat{y}_k + \bar{A}^*_{22}\hat{z}_k + \bar{C}\bar{B}^*_k - \bar{L}\nu_k \\ &= (\bar{A}^*_{21} - \bar{L}\bar{A}^*_{11})\hat{y}_k + (\bar{A}^*_{22} - \bar{L}\bar{A}^*_{12})\hat{z}_k + (\bar{C} - \bar{L}\bar{N}_C)\bar{B}^*_k + \bar{L}y_{k+1}.\end{aligned}$$

This observer is of reduced order. Compared to previous approach there is no chattering problem, as mentioned in [52], neither a need for an auxiliary controller. The simulation results presented in Section 2.4.7 illustrate this.

2.4.6 With nonlinear terms and/or perturbations

One of the main appeals for constructing discontinuous observers is their ability to provide a good estimate even if the dynamics includes bounded nonlinearities and/or uncertainties. We shall now turn our attention to this case. With our notation, we now have $\xi \not\equiv 0$ and it is a function of t and x. Let us denote by $F\colon \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ the map that gives x_{k+1} in terms of the triple (x_k, t_k, t_{k+1}). With an LTI system, we have $F(x_k, t_k, t_{k+1}) = \mathcal{A}^* x_k$, with the difference $t_{k+1} - t_k$ hidden in $\bar{\mathcal{A}}^*$. The measurement from the system is of the form

$$y_{k+1} = \bar{C}F(x_k, t_k, t_{k+1})$$

Using the same procedure as before, (2.4.14) is equivalent to

$$\nu_k = \bar{C}\bar{\mathcal{A}}^* \begin{bmatrix}\hat{y}_k \\ \hat{z}_k\end{bmatrix} + \bar{C}\bar{B}^*_k - \bar{C}F(x_k, t_k, t_{k+1}),$$

which leads to the error dynamics

$$\widetilde{z}_{k+1} = \bar{N}_C(\hat{x}_{k+1} - x_{k+1}) = \bar{N}_C\bar{\mathcal{A}}^*\hat{x}_k - \bar{L}\bar{C}\bar{\mathcal{A}}^*\hat{x}_k + (\bar{L}\bar{C} - \bar{N}_C)F(x_k, t_k, t_{k+1}). \quad (2.4.16)$$

Let us tackle the case where we have a state-independent perturbation term, that is ξ is only a function of t. This additional constraint on ξ gives us a closed-form expression of F, namely

$$F(x_k, t_k, t_{k+1}) = \mathcal{A}^* x_k + p_k \quad (2.4.17)$$

with

$$p_k \coloneqq \int_{t_k}^{t_{k+1}} e^{\mathcal{A}(t_{k+1}-\tau)} B\xi(\tau)\mathrm{d}\tau.$$

We can then go further in the analysis of the error dynamics: using (2.4.17) in (2.4.16) yields

$$\widetilde{z}_{k+1} = (\bar{A}^*_{22} - \bar{L}\bar{A}^*_{12})\widetilde{z}_k + (\bar{L}\bar{C} - \bar{N}_C)p_k$$

One can see that the choice of the gain matrix $\bar{L}$ is affecting both the linear part of the dynamics and how the perturbation affects the error. Hence careful attention has to be given to the choice of $\bar{L}$.

2.4.7 Simulation results

Let us illustrate the effectiveness of the proposed discrete-time sliding observer. We use the same example as in [110], that is a simple pendulum. The ideal dynamics of this system is

$$ml\ddot{\vartheta} + mg \sin \vartheta = 0,$$

which can be put into the state-space form

$$\begin{pmatrix} \dot{x}^{(1)} \\ \dot{x}^{(2)} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix} + \begin{pmatrix} 0 \\ -\sin x^{(1)} \end{pmatrix},$$

with $x^{(1)} = \vartheta$ and $\xi^{(2)} = \dot{\vartheta}$. In the aforementioned reference, the scalar output was chosen as

$$y_k := \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} \hat{x}_k^{(1)} \\ \hat{x}_k^{(2)} \end{pmatrix}$$

The gain matrix is chosen as

$$L := \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The dynamics of the observer is

$$\begin{pmatrix} \hat{x}_{k+1}^{(1)} \\ \hat{x}_{k+1}^{(2)} \end{pmatrix} = \begin{pmatrix} 1 & h/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{x}_k^{(1)} \\ \hat{x}_k^{(2)} \end{pmatrix} + L\nu_k.$$

The non measurable part of the state has dynamics characterized by an eigenvalue at 0.9, inside the unit circle. We compare the performance of this observer with a classical Luenberger one. The dynamics of the latter is

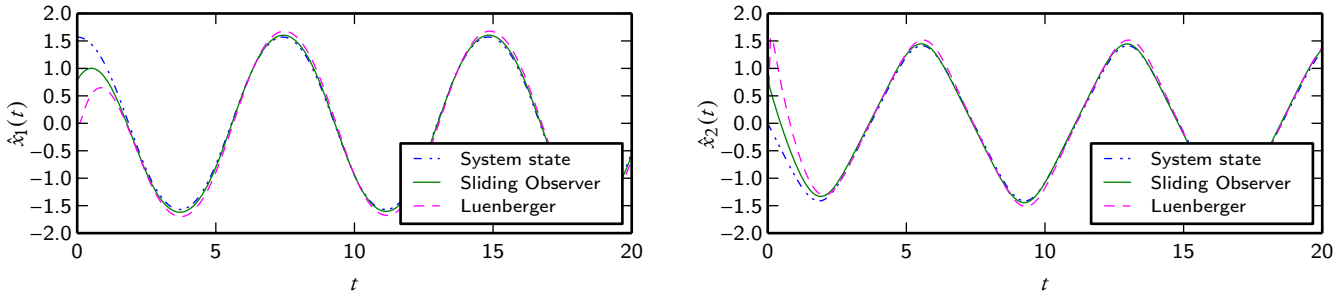
$$\begin{pmatrix} \hat{x}_{k+1}^{(1)} \\ \hat{x}_{k+1}^{(2)} \end{pmatrix} = \begin{pmatrix} 1 & h/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{x}_k^{(1)} \\ \hat{x}_k^{(2)} \end{pmatrix} + L(\hat{y}_k - y_k),$$

with the same gain matrix L. This places the eigenvalues of $A^* + LC$, of the error dynamics, at 0.89 and 0.11, inside the unit circle.

2.5 Perturbation Attenuation Improvements

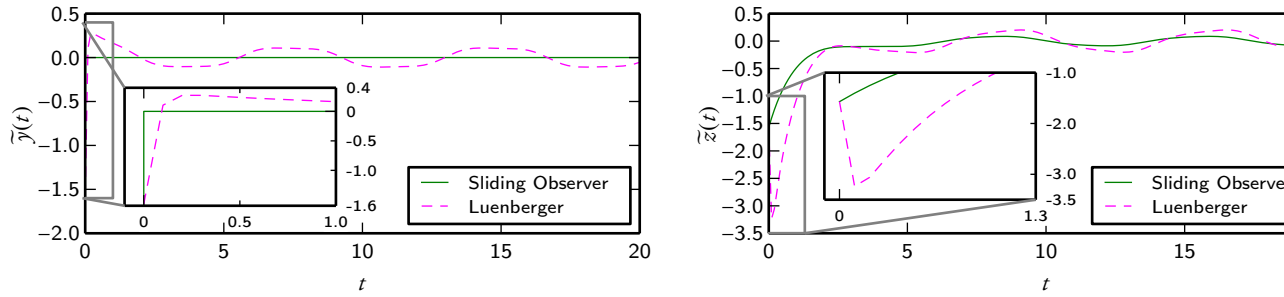
2.5.1 Problem statement

In continuous time, while in the sliding phase, the discontinuous control input u_{cont}^{δ} takes values to reject the perturbation action, according to Filippov's concept of solutions. Let



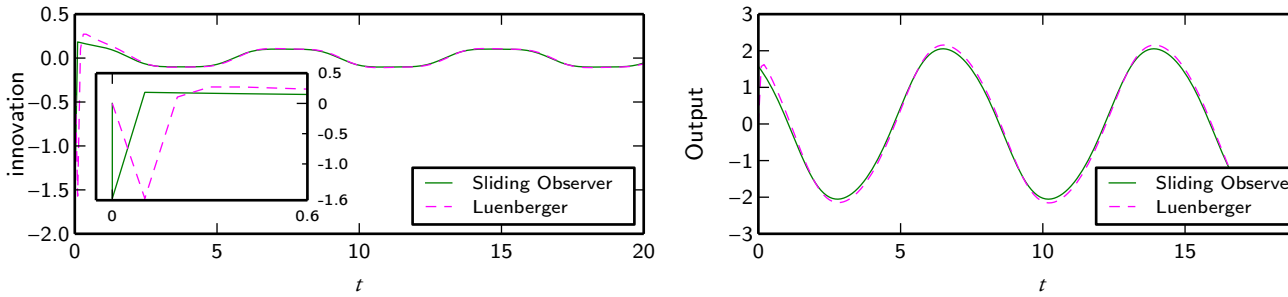
(a) Evolution of x_1 (b) Evolution of x_2

Figure 2.1: Evolution of the system states and the estimates



(a) Evolution of $\tilde{y} = \hat{y} - y$ (b) Evolution of $\tilde{z} = \hat{z} - z$

Figure 2.2: Evolution of the errors between the estimates and the true values



(a) Evolution of the innovation (b) Evolution of $\hat{y}$

Figure 2.3: Innovation and estimated output $\hat{y}$

us illustrate this using an academic example, $\dot{x}(t) \in -\mathrm{Sgn}(x) + d(t)$, x scalar and d a continuous unknown perturbation, with $|d(t)| \leq 1$ for all t. In the sliding phase, the value taken by $\mathrm{Sgn}(x(t))$ is $d(t)$. In Section 2.2, it was established that in the discrete-time sliding phase the input u^s is approximating the opposite of contribution of the perturbation in the evolution of the sliding variable. But this action is a posteriori, in the sense that at time t_k, u^s corrects for the contribution of the perturbation over the time interval $[t_{k-1}, t_k]$.

Hence the attenuation of the perturbation is limited since the control is lagging by one sampling period.

Our approach to enhance the perturbation attenuation consists in using the past values of the sliding variable in order to predict the contribution of the perturbation on the next time interval. Then we take into account this prediction in the control input that is going to be applied for this time interval. As we shall see, this additional control input is computed independently of the other ones. The idea presented here shares some similarities with previous work like [80, 104]

Let us study the dynamics of the ZOH-discretized discrete-time system with a nonzero perturbation. We suppose the system has been in the discrete-time sliding phase for more than 1 sampling period. Adding the perturbation term to (2.2.2), we get

$$x_{k+1} = e^{Ah}x_k + B^* u_k^{eq} + B^* u_k^s + B^* u_k^p + p_k,$$

where the new control input U_k^p will be used to enhance the disturbance compensation. With u_k^{eq} as in (2.2.11), the sliding variable dynamics is

$$\sigma_{k+1} = \sigma_k + CB^*(u_k^s + u_k^p) + Cp_k, \tag{2.5.1}$$

with

$$p_k := \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} B\xi(\tau)\mathrm{d}\tau. \tag{2.5.2}$$

We refer to p_k or Cp_k as the effect, or contribution, of the perturbation. From (2.2.6), in the discrete-time sliding phase $\widetilde{\sigma}_{k+1} = 0$, so $\sigma_k + CB^* u_k^s = 0$. Therefore the relation (2.5.1) becomes

$$\sigma_{k+1} = Cp_k + CB^* u_k^p. \tag{2.5.3}$$

If we have an estimate $\widetilde{Cp}_k$ of Cp_k at the time t_k, then we can use this information in the control input u_k^p, defined as

$$u_k^p := -(CB^*)^{-1}\widetilde{Cp}_k. \tag{2.5.4}$$

Then injecting (2.5.4) in (2.5.3), we get:

$$\sigma_{k+1} = Cp_k - \widetilde{Cp}_k =: C\hat{p}. \tag{2.5.5}$$

The Algorithm 1 summarizes the computation of u_k^p.

Algorithm 1 Computation of u_k^p

Require: vector buf of previous Cp_l
  while $u_{k-1} \notin (-\alpha, \alpha)^p$ do {Reaching phase}
    $u_k^p \leftarrow 0$
  end while
  repeat {Initialize the prediction}
    $u_k^p \leftarrow 0$
    Prepend $\sigma_{k+1} = Cp_{k-1}$ to buf
  until buf is full
  loop {Iteration in the discrete-time sliding phase}
    $Cp_{k-1} \leftarrow \sigma_{k+1} + \widetilde{Cp}_{k-1}$
    Erase the last value of buf and prepend Cp_{k-1} to buf
    Compute $\widetilde{Cp}_k$ and project $\widetilde{Cp}_k$ onto $\mathcal{B}_{\alpha\beta}$
    $u_k^p \leftarrow -(CB^*)^{-1}\widetilde{Cp}_k$
  end loop

In the rest of this note, we name the quantity denoted by $C\hat{p}$ the residual perturbation. The chattering on the output σ is equal to the prediction error on the effect of the perturbation. Therefore to reduce the chattering imputable to perturbations like unmodeled dynamics, we can try to find a good prediction of the value of the perturbation. The better it is, the smaller the chattering will be. In the next section, we build an estimate $\widetilde{Cp}_k$ of Cp_k at the time t_k.

2.5.2 Prediction of the perturbation

The method we propose to make use of is finite difference, itself based on the Taylor expansion of a function. For instance if we suppose that a function $f : \mathbb{R} \to \mathbb{R}^n$ is twice differentiable, it holds that

$$f(x+h) = f(x) + h\dot{f}(x) + \frac{h^2}{2}\ddot{f}(x) + O(h^3).$$

To use this recurrence formula for the prediction at the next time instant, one needs to estimate the values of the first and second derivatives of the function. We use again the finite difference method to get them. We suppose we have access to Cp_k for the previous r sampling periods. However $Cp(t)$ is not a continuous function. It is piecewise smooth, but only right-continuous. From the definition in (2.5.2), we have $Cp(t_k^+) = 0$ for all $k \in \mathbb{N}$.

This prevents us from directly applying the Taylor expansion to $Cp(t)$ itself. Nonetheless, we can use the Taylor expansion formula on the perturbation ξ and then derive the one for the prediction of Cp_k. Let us illustrate this using a second order expansion of ξ:

$$\xi(\tau + h) = \xi(h) + h\dot{\xi}(\tau) + \frac{h^2}{2}\ddot{\xi}(\tau) + O(h^3).$$

Multiplying by $e^{\mathcal{A}(t_k-\tau)}$ and integrating, we get:

$$\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\xi(\tau + h)\mathrm{d}\tau = \int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\xi(\tau)\mathrm{d}\tau +$$
$$h\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\dot{\xi}(\tau)\mathrm{d}\tau + \frac{h^2}{2}\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\ddot{\xi}(\tau)\mathrm{d}\tau + O(h^4). \tag{2.5.6}$$

Using the change of variable $\tau' = \tau + h$ in the left-hand side of (2.5.6) and using (2.5.2), we obtain the relation

$$Cp_k = Cp_{k-1} + hC\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\dot{\xi}(\tau)\mathrm{d}\tau$$
$$+\frac{h^2}{2}C\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)} B\ddot{\xi}(\tau)\mathrm{d}\tau + O(h^4). \tag{2.5.7}$$

Let us define $Cp_k^{(i)} := C\int_{t_{k-1}}^{t_k} e^{\mathcal{A}(t_k-\tau)}\xi^{(i)}(\tau)\mathrm{d}\tau$. We use the finite difference method to estimate the second and third terms in (2.5.7).

Lemma 2.5.1. *Suppose we have an r-step approximation of the i-th derivative of ξ, $\xi^{(i)}(\tau) = \sum_{l=0}^{r}\alpha_l\xi(\tau - lh) + O(h^{r-i+1})$ with $r \geq i$. Then the approximation formula for $Cp_k^{(i)}$ is $\sum_{l=0}^{r}\alpha_l Cp_{k-l}$ and is of order $O(h^{r-i+2})$.*

Proof. Starting from the approximation relation for the derivative and with basic operations, we get:

$$C\int_{t_k}^{t_{k+1}} e^{\mathcal{A}(t_{k+1}-\tau)} B\xi^{(i)}(\tau)\mathrm{d}\tau =$$
$$\sum_{l=0}^{r}\alpha_l C\int_{t_k}^{t_{k+1}} e^{\mathcal{A}(t_{k+1}-\tau)} B\xi(\tau - lh)\mathrm{d}\tau + O(h^{r-i+2}).$$

For each integral, we use the change of variable $\tau' = \tau - lh$.

$$C\int_{t_k}^{t_{k+1}} e^{\mathcal{A}(t_{k+1}-\tau)} B\xi^{(i)}(\tau) =$$

$$\sum_{l=0}^{r} \alpha_l C \int_{t_{k-l}}^{t_{k+1-l}} e^{\mathcal{A}(t_{k+1-l}-\tau')} B\xi(\tau')\mathrm{d}\tau' + O(h^{r-i+2})$$
$$= \sum_{l=0}^{r} \alpha_l C p_{k-l} + O(h^{r-i+2}).$$

□

Using Lemma 2.5.1 for the approximate values of the first and second derivatives in (2.5.7) yields:

$$Cp_k \approx Cp_{k-1} + h \sum_{l=0}^{r_1} \alpha_l Cp_{k-l} + \frac{h^2}{2} \sum_{l=0}^{r_2} \alpha'_l Cp_{k-l}. \tag{2.5.8}$$

where r_1 and r_2 (both $\leq r$) are the orders for the prediction of the first and second derivatives. Those, with the coefficients α_l and α'_l, are taken from [40]. As Equation (2.5.8) illustrates it, the i-th derivative estimate is multiplied by h^i. By Lemma 2.5.1 we know that the approximation error for the i-th derivative is of order $O(h^{r-i+2})$. Then the approximation error introduced by the i-th derivative is of order $O(h^{r+2})$.

Let us present some possible prediction formulæ. Using only the first derivative, we have the relation

$$\begin{aligned} Cp_k &= 2Cp_{k-1} - Cp_{k-2} + O(h^3) \\ &= \widetilde{Cp}_k + O(h^3), \end{aligned} \tag{2.5.9}$$

where both the approximation of the first derivative and the prediction have the same order. Adding the second derivative yields

$$\begin{aligned} Cp_k &= \frac{5}{2} Cp_{k-1} - 2Cp_{k-2} + \frac{1}{2} Cp_{k-3} + O(h^3) \\ &= \widetilde{Cp}_k + O(h^3). \end{aligned} \tag{2.5.10}$$

The order is the same as in (2.5.9) since we use a 1-step approximation for $Cp_k^{(1)}$, of order 1 which introduces an error of order $O(h^3)$ in (2.5.10). Therefore to achieve an overall error of order $O(h^4)$, we need an estimate with an order 2 for $Cp_k^{(1)}$, yielding

$$\begin{aligned} Cp_k &= 3Cp_{k-1} - 3Cp_{k-2} + Cp_{k-3} + O(h^4) \\ &= \widetilde{Cp}_k + O(h^4). \end{aligned} \tag{2.5.11}$$

From now on, let us refer to (2.5.9) as a linear or order 1 prediction and to (2.5.10) as an order 2 prediction. Both predictions are said to be high-order in contrast with the one given by the relation

$$\widetilde{Cp_k} = Cp_{k-1}, \tag{2.5.12}$$

which is said to be constant. Note that the order of the prediction indicates also the number of times the perturbation $\xi(t)$ has to be differentiable for the approximation order to be guaranteed.

2.5.3 Controller implementation

Let $p \in \mathbb{N}$ be the approximation order. As long as the closed-loop system is not in the discrete-time sliding phase, we set $u^p \equiv 0$. Let $k_0 \in \mathbb{N}$ be such that $\widetilde{\sigma}_{k_0} = 0$, that is the system is in the discrete-time sliding phase, and let $k > k_0$. For $k_0 \leq k < k_0 + p$, we let $u_k^p = 0$ and we save σ_k, which is Cp_{k-1} according to (2.5.3). For $k \geq p + k_0$, we compute the prediction of the perturbation $\widetilde{Cp}_k$ using one of the formulæin (2.5.9), (2.5.11) or (2.5.12) and we set $u_k^p = (CB)^{-1}\widetilde{Cp}_k$. At this point, we cannot know Cp_{k-1} directly from σ_k since u_k^p changes the dynamics. From (2.5.5), at time t_k, Cp_{k-1} is obtained through the relation

$$Cp_{k-1} = \sigma_k + \widetilde{Cp}_{k-1},$$

where σ_k is measured and $\widetilde{Cp}_{k-1}$ is known from the last estimation.

In Proposition 2.2.11, the condition $\|Cp_k\| \leq \alpha\beta$ for all k, with β the smallest eigenvalue of $CB_s^* := (CB + (CB^*)^T)/2$, was used to guarantee a finite-time reaching phase and that the system does not exit the discrete-time sliding phase. This inequality condition $\|Cp_k\| < \alpha\beta$ gives us a "feasibility set" for the prediction $\widetilde{Cp}_k$. If $\widetilde{Cp}_k > \alpha\beta$, then we project the estimate onto the ball of radius $\alpha\beta$ centered at the origin. With this refinement of our algorithm, we make a first step towards some stability properties, which are studied in Section 2.5.6.

2.5.4 Numerical example

We illustrate the method on a simple 2 dimensional system, that we use again in Section 3.4:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B\xi(t) & A = \begin{pmatrix} 0 & 1 \\ 19 & -2 \end{pmatrix}, \\ \sigma = Cx & \\ u(t) = u^{eq}(t) + u^s(t) & B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ C^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{cases} \tag{2.5.13}$$

The perturbation ξ is a simple sinusoid: $\xi(t) = \sin(4\pi t)$. It was chosen as an example of unmodeled dynamics in a mechanical system. The matrix A has the eigenvalues $\lambda_1 = 3.47$ and $\lambda_2 = -5.47$. The dynamics on the sliding surface is given by $\begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}$, which has

eigenvalues 0 and -1. The initial state is $(-15, 20)^T$ and the sampling period is 10^{-2}s. Simulation have been done with 4 different controllers: the classical implicit one, one with the assumption that the perturbation is constant (2.5.12), one with the prediction presented in (2.5.9) and the last one with the prediction formula in (2.5.11). The simulations run for 150 s and were carried out with the SICONOS software package [2][1]. Figures were created using Matplotlib [62]. We present plots of the state variables in Figures 2.1, 2.2 and 2.3. Then we display the evolution of the discontinuous control u_s in Figures 2.4 and 2.5. On each figure depicting the state space evolution, the black line is the hyperplane $\sigma = 0$.

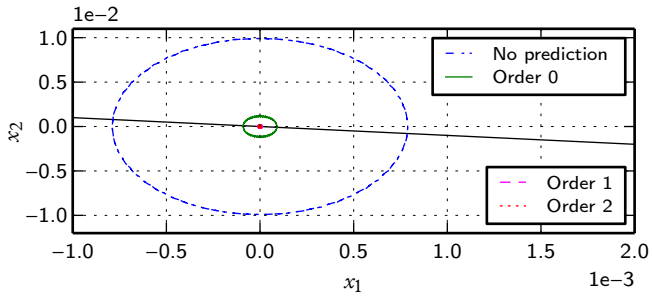


Figure 2.1: Simulations of system (2.5.13), zoomed around the origin, with $h = 10^{-2}$s.

In Figure 2.1, the state values for the last second are displayed, when each system features some kind of steady-state behaviour. The improvement obtained using the prediction is clearly visible, to the point that we need to zoom to see the chattering with higher-order estimations. In Figure 2.2, detail of Figure 2.1, only the trajectories of the closed-loop

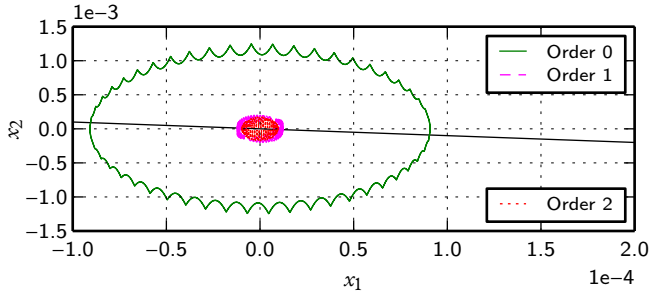


Figure 2.2: Detail of Figure 2.1, with only the trajectories of the closed-loop systems with prediction visible.

systems with prediction are displayed. The behaviour between sampling times can be seen with the constant prediction (order 0). Each high-order prediction (order 1 and 2),

[1] http://siconos.gforge.inria.fr

yields a closed-loop system featuring even less chattering, as we can see in Figure 2.3. At

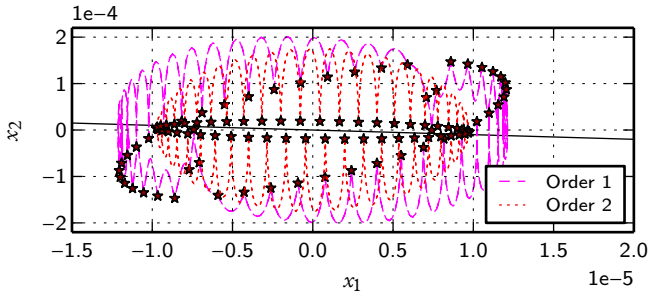


Figure 2.3: Detail of Figure 2.2, with only the trajectories of the closed-loop system with higher-order prediction visible.

this level of detail, the difference between the two estimations with high-order becomes visible. Markers indicate the state of the system at each time instant t_k, that is when the control values change. We see that for those particular values, the prediction with order 2 yields better results, with a chattering one order of magnitude smaller. However with the small contribution of the residual perturbation, the inter-sampling dynamics provides an important contribution to the behaviour of the system. For the present example, the advantages of using a prediction order higher than 1 are not so striking as the ones from using at least a linear prediction. Therefore it is important to take into account the inter-sampling dynamics when designing the prediction of the perturbation.

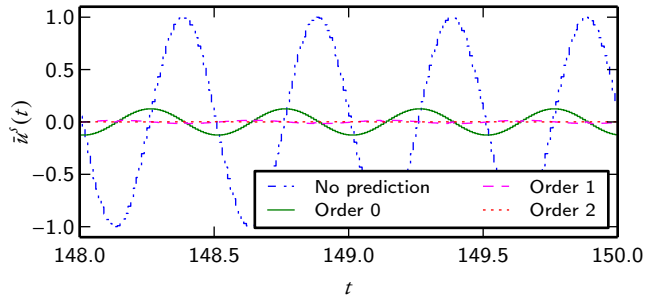


Figure 2.4: Evolution of u^s for simulations of system (2.5.13) with $h = 10^{-2}$s.

Let us turn our attention to the control input. Firstly we plot the evolution of the discontinuous part u^s, which is, up to a constant, equal to the residual perturbation (2.5.5). In Figure 2.4, the evolution of u^s for the last 2 seconds is displayed. As expected, the better the approximation, the smaller u^s is. In Figure 2.5, detail of Figure 2.4, the same phenomenon can be seen for the high-order estimations. Let us turn our attention to u^p, as displayed in Figure 2.6. In Section 2.2, it is shown that with no perturbation prediction, u^s

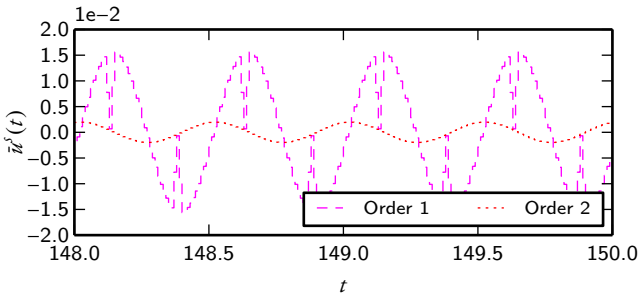


Figure 2.5: Detail of Figure 2.4.

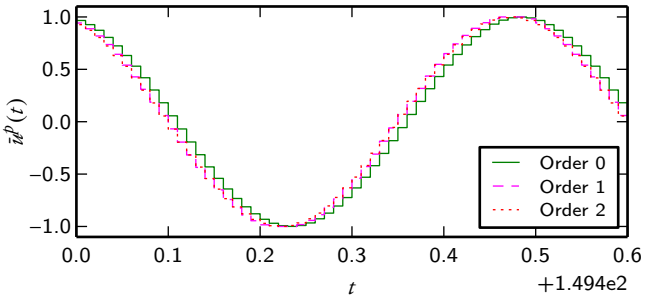


Figure 2.6: Evolution of u^p for simulations of system (2.5.13)

approximates the perturbation with a delay of h. Recall that with an order 0, the predicted value of the effect of the perturbation is the one the system just measured. In this case, the values taken by u^p are the ones that u^s would take with no perturbation prediction. What we can see from Figures 2.6 and 2.7 is that with a high-order prediction, this lag of one sampling period h seems to have vanished. This is easier to assess on Figure 2.7, where if we shift the solid green curve (corresponding with the constant prediction) by $-h$ on the time axis, then it would more or less overlap with the two other curves. This observation can also be explained in the following way: with an estimation of order 0, there is no use of a derivative of ξ to "look into the future". When this is the case, as for the order 1 and 2, the main difference between the different predictions is the value that u^p_k takes. This highlights the fact that using high-order estimation yields a prediction, hence a chattering attenuation, substantially better than a constant one. A last interesting comparison is the control effort used in each of these closed-loop systems. In particular, we want to compare the sum $u^s + u^p$ with the discontinuous input u^s in the case where $u^p \equiv 0$. In Figure 2.8, one can see that all the control inputs are quite close, in terms of shape and value. At this level of detail, there is no difference between the closed-loop system with prediction. The main difference is that without prediction, the control is shifted by h with respect to the one with a prediction. Adding a perturbation prediction yields no extra control effort in

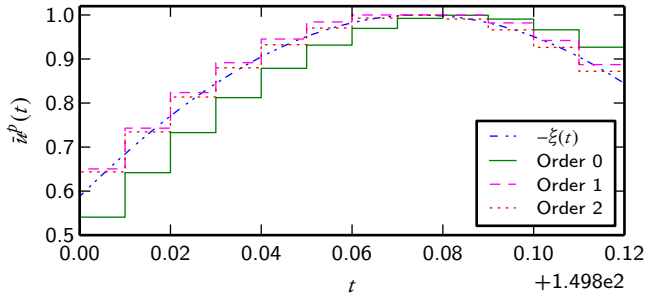


Figure 2.7: Detail of Figure 2.6.

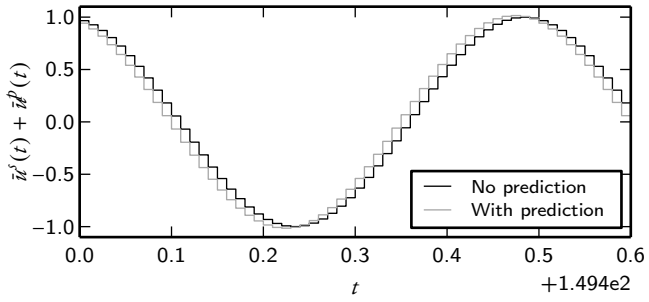


Figure 2.8: Evolution of $u^s + u^p$ for simulations of system (2.5.13)

this example.

2.5.5 Multirate sampling

As we have seen in Section 2.5.2, the prediction process consists in two stages: first the different derivatives are approximated and then the prediction is made. The same sampling period h was used for both stages. In the following we change the sampling period used for the estimation of the coefficients. We assume that we can sample the output of the system more often than we can change the control. Let us denote by $h_s > 0$ the sampling time interval for the state of the system. To simplify the analysis, we assume that the ratio between h and h_s is an integer s. We also need to measure the whole state. Previously, it was sufficient to be able to measure the sliding variable. However this requirement is not too restrictive since for the computation, it is generally required to measure the full state since it is needed for the computation of the equivalent part of the control input. Let us define the quantity $p_{k+\frac{l}{s}} := \int_{t_k}^{t_k+\frac{l}{s}h} e^{(t_k+\frac{l}{s}h-\tau)}\xi(\tau)\mathrm{d}\tau$. During the first step of the algorithm, we have to be able to compute the derivative up to a certain order using the available measured variables. The following lemma contains a formula relating the two quantities.

Lemma 2.5.2. *Suppose that we have an estimate of the i-th derivative of ξ: $\xi^{(i)}(t_k)$ =*

$\sum_{l=1}^{s} \alpha_l \xi(t_k - \frac{lh}{s})$. *Then the approximation formula for* $Cp_k^{(i)}$ *is*

$$\sum_{l=1}^{s} \alpha_l Cp_{k+\frac{s-l}{s}} + \sum_{l=1}^{s} \alpha_l e^{A\frac{(s-l)h}{s}} Cp_{k-1} - e^{Ah} \sum_{l=1}^{s} \alpha_l Cp_{k-1+(s-l)/s}$$

Proof. As in the proof of Lemma 2.5.1, we consider the estimate formula and transform it to obtain the desired quantity:

$$\int_{t_k}^{t_k+1} e^{A(t_{k+1})} B\xi^{(i)}(t_k) = \sum_{l=1}^{s} \alpha_l \int_{t_k}^{t_{k+1}} e^{A(t_{k+1})} B\xi(\tau - lh_s)\mathrm{d}\tau. \tag{2.5.14}$$

Let us study each element of the sum separately. We apply the change of variable $\tau' = \tau - lh/s = \tau - lh_s$:

$$\begin{aligned}
\int_{t_k}^{t_{k+1}} e^{A(t_{k+1})} B\xi(\tau - lh_s)\mathrm{d}\tau &= \int_{t_k - lh_s}^{t_{k+1}-lh_s} e^{A(t_{k+1}-lh_s-\tau')} B\xi(\tau')\mathrm{d}\tau' \\
&= Cp_{k+\frac{s-l}{s}} + \int_{t_k-lh_s}^{t_k} e^{A(t_k+(s-l)h_s-\tau')} B\xi(\tau')\mathrm{d}\tau' \\
&= Cp_{k+\frac{s-l}{s}} + e^{A(s-l)h_s} \int_{t_k-\frac{lh}{s}}^{t_k} e^{A(t_k-\tau')} B\xi(\tau')\mathrm{d}\tau' \\
&= Cp_{k+\frac{s-l}{s}} + e^{A(s-l)h_s} \left(\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau')} B\xi(\tau')\mathrm{d}\tau' \right. \\
&\qquad \left. - \int_{t_{k-1}}^{t_{k-1+\frac{(s-l)h}{s}}} e^{A(t_k-\tau')} B\xi(\tau')\mathrm{d}\tau' \right) \\
&= Cp_{k+\frac{s-l}{s}} + e^{A(s-l)h_s} \left(Cp_{k-1} - e^{Alh_s} Cp_{k-1+\frac{s-l}{s}} \right) \\
&= Cp_{k+\frac{s-l}{s}} + e^{A(s-l)h_s} Cp_{k-1} - e^{Ah} Cp_{k-1+\frac{s-l}{s}}.
\end{aligned}$$

Then using this expression in (2.5.14) yields:

$$\begin{aligned}
\int_{t_k}^{t_k+1} e^{A(t_{k+1})} B\xi^{(i)}(t_k) &= \sum_{l=1}^{s} \alpha_l Cp_{k+\frac{s-l}{s}} \\
+ \sum_{l=1}^{s} \alpha_l e^{A\frac{(s-l)h}{s}} Cp_{k-1} - e^{Ah} \sum_{l=1}^{s} \alpha_l Cp_{k-1+(s-l)/s}&.
\end{aligned}$$

Using this expression back in the original approximation formula completes the proof. □

The prediction step has the same sampling period h as before. However it might be interesting to use a higher-order extrapolation formula: the estimates of the "derivatives" $Cp_k^{(i)}$ are know with a higher precision since h_s is supposed the be smaller than h.

2.5.6 Stability properties

One natural question arising from this prediction procedure is whether the use of a bad prediction can degrade the closed-loop performance instead of improving it. Sliding Mode Control is known for its robustness and such a feature should not be lost while trying to reduce the chattering.

First we characterize the robustness with a discrete-time feedback loop. Since we use an implicit discretization of the discontinuous control u^s, we can divide the evolution of the closed-loop system in two phases: the reaching phase, were u^s has at least one of its element taking the maximum value $\pm\alpha$ and the discrete-time sliding phase from Definition 2.2.4. We define the robustness as the feature that the system does not leave the discrete-time sliding phase, once it enters it.

Proposition 2.5.3. *Suppose that CB_s^* is positive-definite and $\beta > 0$ is its smallest eigenvalue. Let the estimate $\widetilde{Cp}_k$ be obtained by the mean of one of the formulæin Section 2.5.2. If $\alpha > 0$ is such that for all $k \in \mathbb{N}$, $2\|Cp_k\| < \alpha\beta$ and $\widetilde{Cp}_k$ is projected onto the admissible set $\{v \in \mathbb{R}^p : \|v\| < \alpha\beta\}$, then the perturbed closed-loop system given by* (2.5.1) *and* (2.2.6) *enters the discrete-time sliding phase in finite time, and stays in it.*

Proof. Let $V(\sigma_k) := -u_{k-1}^{sT}\sigma_k$, $u_{k-1}^s \in -\alpha\,\mathrm{Sgn}(\sigma_k)$, be the Lyapunov function candidate. The function $V(\,\cdot\,)$ is positive definite, radially unbounded, and decrescent since $V(\sigma_k) = \alpha\|\sigma_k\|_1^2$ and $\alpha > 0$. Assume that the system is initialized outside the discrete-time sliding phase, and recall that in the reaching phase, $u^p \equiv 0$. From Definition 2.2.4, it follows that $\|u_k^s\| \geq \alpha$. Let us study the variations of $V(\,\cdot\,)$:

$$\begin{aligned} V(\sigma_{k+1}) - V(\sigma_k) &= -(u_k^s)^T\sigma_{k+1} + (u_{k-1}^s)^T\sigma_k \\ &= -(u_k^s)^T(\sigma_k + CB^*u_k^s + Cp_k) + (u_{k-1}^s)^T\sigma_k \\ &= -(u_k^s)^TCB^*u_k^s + (u_k^s)^TCp_k + \langle u_{k-1}^s - u_k^s, \sigma_k\rangle. \end{aligned}$$

The inclusion $-u_k^s \in \alpha\,\mathrm{Sgn}(\sigma_{k+1})$ is equivalent to $\sigma_{k+1} \in N_{\alpha\mathcal{B}_\infty}(-u_k^s)$. This implies that the last term is always nonpositive by the definition of the normal cone. Therefore we have $V(\sigma_{k+1}) - V(\sigma_k) \leq -\beta\|u_k^s\|^2 - (u_k^s)^TCp_k$. Using the Cauchy-Schwarz inequality, we obtain $|(u_k^s)^TCp_k| \leq \|u_k^s\|\|Cp_k\|$. To ensure that $V(\,\cdot\,)$ decreases strictly, we need $\|Cp_k\| < \beta\|u_k^s\|$. This condition is satisfied using the hypothesis on the gain α and the fact that $\beta > 0$. Note that even in the case with multiple switching surfaces, $V(\,\cdot\,)$ decreases as long as the system is not "sliding" on the intersection of all the manifolds. If $\widetilde{\sigma}_{k+1} = 0$, then we enter the discrete-time sliding phase. Let $\varkappa = \alpha\beta - \|Cp_k\|$. From the assumption,

$\varkappa > 0$ holds. In the reaching phase, $V(\cdot)$ decreases by at least $\varkappa\alpha$ at each sampling period. Hence, $V(\sigma_k)$ converges to 0 in finite-time.

Let the system be in the discrete-time sliding phase at t_k. Then from (2.2.6) and (2.5.4) $\widetilde{\sigma}_{k+1} = 0$, $u_k^p = -(CB^*)^{-1}\widetilde{Cp}_k$ and $\sigma_{k+1} = C\hat{p}_k$. At time t_{k+1}, we have $\widetilde{\sigma}_{k+2} = C\hat{p}_k + CB^*u_{k+1}^s + CB^*u_{k+1}^p$. Let us show that $u_{k+1}^s = -(CB^*)^{-1}C\hat{p}_k$ is the unique solution to the generalized equation (2.2.6). With this value, $\widetilde{\sigma}_{k+2} = 0$. Note that $\|C\hat{p}_k\| \leq 2\|Cp_k\|$ since we project the estimate $\widetilde{Cp}_k$. Using Lemma 2.2.10 and the hypothesis $2\|Cp_k\| < \alpha\beta$, we get $\|u_{k+1}^s\| \leq \beta^{-1}\|C\hat{p}_k\| < \alpha$. Relations between norms yield $\|u_{k+1}^s\|_\infty < \alpha$. Then $u_{k+1}^s \in (\alpha, \alpha)^p \subset \alpha\,\mathrm{Sgn}(0)$ and u_{k+1}^s is a solution to (2.2.6). Remember from Lemma 2.2.3 that this solution is unique since CB^* is positive-definite. Thus $u_{k+1}^s = -(CB^*)^{-1}C\hat{p}_k$ is the unique solution to (2.2.6) at time t_{k+1}, and by induction, the system stays in the discrete-time sliding phase. □

The closed-loop system does not exit the discrete-time sliding surface, even if the estimate is quite far from the actual value. In the worst case the magnitude of the chattering is doubled. The only change required by the prediction is to double the bound on the discontinuous control input u^s. This is possible without degrading the performance with the implicit discretization of u^s, since the numerical chattering is non-existent with a LTI system discretized using ZOH.

Remark 2.5.4. If it is not possible to double the bound α, it is possible to project the estimate onto a ball with a smaller radius $\gamma\alpha\beta$, where $(1+\gamma)\alpha$ is less than the componentwise upper bound on the control input. It is then easy to adapt the previous proposition to have a result with this estimate. However the chattering reduction might be less appealing.

Conclusion

This chapter begins with the study of the discretization of the ECB-SMC controller. After a quick presentation of the different discrete-time controllers, we focus on the ones that feature an implicitly discretized $\mathrm{Sgn}(\cdot)$ multifunction. Several properties of this sliding mode controller are provided: Lyapunov stability of the discrete-time sliding variable, finite-time reachability of the sliding surface, convergence of the discrete-time control input to the continuous-time one and perturbation attenuation. Those results, mainly for the sliding variable, are valid for the closed-loop system using a special discrete-time sliding mode control scheme on the equivalent part. When the latter is discretization in another

way, this correspondence fails to hold. We also consider several time discretizations of the classical ECB-SMC controller and underline the influence of the discretization method for the state-continuous part of the input. This topic is again tackled in Section 3.4: we provide an example where the use of an explicit method can make the closed-loop system diverge, whereas with the other methods it reaches the sliding surface. Finally we provide a performance analysis of the different discretization methods, which highlights the ability of the implicit one to alleviate or suppress the numerical chattering. The implicit discretization method is also applied on the twisting algorithm. However, it turns out that the origin can only be reached from very specific initial positions. Subsequently, the structure of the controller is modified by considering the underlying AVI. With this modification, the finite-time Lyapunov stability for simple dynamics can be shown. We touch upon the topic of observation by studying some simple sliding mode observers. Finally, a modified discrete-time sliding mode controller is proposed, which can improve the perturbation attenuation of smooth perturbations. The core idea is to use the previous values of the sliding variable to provide an estimate of the effect of the perturbation on the system for the next sampling period. The method proposed here is adapted from finite-difference formulæ. Stability properties of the resulting controller are also discussed.

Chapter 3

Simulations of Sliding Mode Controllers

The simulation of the control loop in an important step in the process of designing a controller. It offers a fast and competive way to test a design, which enables fast development cycles. In the fields of control theory, the pair MATLAB®–Simulink® is almost the only tool used to perform simulations. Extensions known as toolboxes can extend the functionalities such as providing a way to translate a Simulink® model into C code, in order to run the control loop on an embedded controller. However, MATLAB®–Simulink® has some weaknesses, in particular in the numerical integration. It is not possible to integrate parts of a dynamics in a fully implicit way. The results from the previous chapter underline that it is paramount to implicitly discretize the argument of the Sgn multifunction. This prompted us to disregard this platform for simulation purposes. Instead we developed a Control toolbox in the INRIA SICONOS platform and simulated control loops using this software.

In this chapter we first present some algorithm to compute the control input value when it is the solution of an AVI. Then we move to the topic of the computation of the control input with nonlinear dynamics. Section 3.3 is dedicated to a presentation of the SICONOS platform with a particular attention given to the work done by the author. The last two sections present some simulation results: the first one is an analysis from a numerical point of view of the various discretization schemes introduced in Section 2.2. The last section presents some simulation results illustrating the benefits from using the method proposed in Section 3.2.

3.1 Solvers for the Computation of the Control Input

This section is dedicated to the computation of a solution to the AVI arising in the discrete-time SMC case. We shall proceed here from the simplest case to the most general one. But first, let us recall the task at hand: with p the dimension of the control input, the AVI(K, q, M) consists in the following: given a polyhedral set K, a vector $q \in \mathbb{R}^p$ and matrix $M \in \mathbb{R}^{p\times p}$, we want to find $\lambda \in K$ such that

$$\langle y - \lambda, q + M\lambda \rangle \geq 0, \quad \forall y \in K,$$

or equivalently in terms of a generalized equation

$$0 \in q + M\lambda + N_K(\lambda). \tag{3.1.1}$$

In our case, we consider that K is a closed bounded convex polytope.

SCALAR CASE ($P = 1$) The set K is then an interval $[a, b]$, $a, b \in \mathbb{R}$. The control input can be computed as a simple projection onto K:

$$\lambda = -\Pi_K(q/M), \tag{3.1.2}$$

with Π the projection mapping. This formula is justified later on.

DIAGONAL POSITIVE DEFINITE M WITH K A BOX By box we refer to a set defined as the Cartesian product of intervals: $K = \prod_i I_i$. It can also be seen as a scaled ball for the maximum norm. In this case, the same method as in the scalar case can be applied component-wise:

$$\lambda_i = -\Pi_{I_i}(q_i/m_{ii}) \quad \forall i \in \{1, \dots, n\}. \tag{3.1.3}$$

This expression is a special case of the next method.

SYMMETRIC POSITIVE DEFINITE M We develop here a projection algorithm which takes advantage of the structure of M. Recall from Section 2.1.4 that a symmetric positive definite matrix admits a square root: there exists $R = R^T > 0$ such that $R^T R = M$. Starting from the generalized equation (3.1.1), we use the transformation $\hat{\lambda} = R\lambda$:

$$\begin{aligned} 0 &\in R^T\hat{\lambda} + q + N_K(R^{-1}\hat{\lambda}) \\ 0 &\in \hat{\lambda} + R^{-T}q + R^{-T}N_K(R^{-1}\hat{\lambda}). \end{aligned} \tag{3.1.4}$$

Recalling that $N_K = \partial\delta_K$ and using the chain rule for subdifferential given in Theorem A.1.15, we get

$$\partial\left(\delta_K \circ R^{-1}\right)(\hat{\lambda}) = R^{-T} N_K(R^{-1}\hat{\lambda}).$$

Note that if $R^{-1}\hat{\lambda} \in K$, then $\hat{\lambda} \in \hat{K}$, with

$$\hat{K} := \left\{x \in \mathbb{R}^p \mid \mathcal{A}_K R^{-1} x \geq B_K r\right\} \text{ if } K = \{x \in \mathbb{R}^p \mid \mathcal{A}_K x \geq B_K\}. \tag{3.1.5}$$

Whence it holds that $\left(\delta_K \circ R^{-1}\right)(\hat{\lambda}) = \delta_{\hat{K}}(\hat{\lambda})$ and therefore $N_{\hat{K}}(\hat{\lambda}) = R^{-T} N_K(R^{-1}\hat{\lambda})$. This identity enables us to transform (3.1.4) into

$$0 \in \left(I + N_{\hat{K}}\right)\hat{\lambda} + R^{-T} q.$$

Using Proposition A.1.19 we get the identity $\Pi_{\hat{K}} = (I + N_{\hat{K}})^{-1}$. Therefore, the solution is obtain as

$$\hat{\lambda} = \Pi_{\hat{K}}(-R^{-T} q),$$

which in the original basis is given by

$$\lambda = R^{-1}\Pi_{\hat{K}}\left(-R^{-T} q\right). \tag{3.1.6}$$

We shall now specialize this result with $\mathcal{M}$ a diagonal positive definite matrix and K a box to derive the expression provided earlier in (3.1.2) and (3.1.3). In this case, R is also a diagonal positive definite matrix and therefore $\hat{K}$ is also box-shaped as we shall see. If we rewrite (3.1.5) for a box-shaped set K, we get

$$K = \left\{x \in \mathbb{R}^p \mid \begin{pmatrix} I \\ -I \end{pmatrix} x \geq \begin{pmatrix} a \\ -b \end{pmatrix}\right\},$$

and letting $D := \operatorname{diag}\left\{\sqrt{m_{ii}}\right\}$ the square root of $\mathcal{M}$, we have from (3.1.5)

$$\hat{K} = \left\{x \in \mathbb{R}^p \mid \begin{pmatrix} D^{-1} \\ -D^{-1} \end{pmatrix} x \geq \begin{pmatrix} a \\ -b \end{pmatrix}\right\},$$

which can be rewritten as

$$\hat{K} = \left\{x \in \mathbb{R}^p \mid \begin{pmatrix} I \\ -I \end{pmatrix} x \geq \begin{pmatrix} Da \\ -Db \end{pmatrix}\right\}.$$

We can then write $\hat{K} = \prod_i \hat{I}_i$ with $\hat{I}_i = [d_{ii}a_i, d_{ii}b_i]$. With a box-shaped set, the operator $\Pi_{\hat{K}}$ is componentwise defined as

$$\left(\Pi_{\hat{K}}(x)\right)_i = \Pi_{\hat{I}_i}(x_i).$$

Now from (3.1.6), the value of λ_i is defined as

$$\lambda_i = d_{ii}^{-1}\Pi_{\hat{I}}\left(-d_{ii}^{-1}q_i\right) = \begin{cases} -d_{ii}^{-1}d_{ii}a_i = -a_i & \text{if } q_i < a_i \\ -d_{ii}^{-1}d_{ii}b_i = -b_i & \text{if } q_i > b_i \\ -d_{ii}^{-1}d_{ii}^{-1}q_i = -q_i/m_{ii} & \text{if } a_i < q_i < b_i \end{cases}$$

Therefore, we infer that

$$\lambda_i = \Pi_{I_i}\left(-q_i/m_{ii}\right),$$

which is the relation we sought.

GENERIC M In this case, we opt to use the algorithm proposed in [24], which finds a solution whenever the set K is a bounded convex polytope, as mentioned in the last paragraph of Section 4 of this reference. This method realizes a scheme given in [32], which is also a warmly recommended reading if one wants to understand the algorithm. Further references include the PhD theses [23, 77]. The former contains additional informations with respect to the article and in the latter, a variant of the algorithm better suited to sparse problems is proposed. We aim here at providing a description from an implementation viewpoint rather then a theoretical one which is given in the aforementioned references. We shall present a simplified version as implemented by the author in SICONOS, and that is used for the simulations presented in this thesis. This algorithm has different stages, in which the problem is transformed and finally solved. We describe only the last two: the transformation of the AVI into an LCP-like problem, and the procedure to solve this problem using a method similar to the well-known Lemke scheme for LCP. In what follows, we assume that the set K is convex bounded and given as

$$K := \{x \in \mathbb{R}^p \mid Hx \geq b\} \quad \text{with} \quad H \in \mathbb{R}^{m\times p},\, b \in \mathbb{R}^m,\, m > p \tag{3.1.7}$$

To build the LCP-like problem, we need a vertex v of K which can be found by solving a simple linear program (LP) with the constraint that the solution is in K. With v is associated a set of independent active constraints $\mathscr{A}$ such that the square matrix $H_{\mathscr{A}}$, which consists of the rows associated with the active constraints, is full rank. Let $\mathscr{I}$ be the complement of $\mathscr{A}$, that is the set of inactive constraints. Then we compute the following quantities:

$$\overline{M} = \begin{pmatrix} -H_{\mathscr{A}}^{-T}H_{\mathscr{I}}^{T} & H_{\mathscr{A}}^{-T}MH_{\mathscr{A}}^{-1} \\ 0 & H_{\mathscr{I}}H_{\mathscr{A}}^{-1} \end{pmatrix} \quad \text{and} \quad \bar{q} = \begin{pmatrix} H_{\mathscr{A}}^{-T}MH_{\mathscr{A}}^{-1}b_{\mathscr{A}} + H_{\mathscr{A}}^{-T}q \\ H_{\mathscr{I}}H_{\mathscr{A}}^{-1}b_{\mathscr{A}} - b_{\mathscr{I}} \end{pmatrix}. \tag{3.1.8}$$

Now the method used to solve the LCP-like problem is to augment the problem with an auxiliary variable μ and a covering vector d:

$$w = \bar{q} + \overline{M}z + \mu d \quad 0 \leq z \perp w \geq 0 \quad \text{with} \quad z, w, d \in \mathbb{R}^m. \tag{3.1.9}$$

The term μd is here to ensure that by choosing μ big enough, we can start the method with $w \geq 0$ and $z = 0$. The transformation of the AVI into this complementarity problem leads to the following definitions:

$$w = \begin{pmatrix} r_{\mathscr{A}} \\ s_{\mathscr{I}} \end{pmatrix}, z = \begin{pmatrix} r_{\mathscr{I}} \\ s_{\mathscr{A}} \end{pmatrix}, d = \begin{pmatrix} \mathbb{1} \\ 0 \end{pmatrix}$$

$$\text{with the initial values} \quad r_{\mathscr{A}} \geq 0, s_{\mathscr{I}} \geq 0, r_{\mathscr{I}} = 0, s_{\mathscr{A}} = 0 \text{ and } \mu \geq 0.$$

The variable s can be seen as a slack variable for the inequalities defining the set K in (3.1.7): $Hx - s = b$ for all $x \in K$. The vector r is related to the normal cone operator of K: at a point $x \in K$ an element of the normal cone $N_K(x)$ is given by $-H^T r$. Then we have for all i: if $H_{i\bullet}x = b_i$ then $r_i \geq 0$, otherwise $r_i = 0$. It is easy to see that the two variables are related by complementarity: $0 \leq r \perp s \geq 0$. A solution to the AVI is recovered as

$$\lambda = H_{\mathscr{A}}^{-1}(s_{\mathscr{A}} + b_{\mathscr{A}}) \tag{3.1.10}$$

Rather deriving the transformation from the AVI to this LCP-like problem, let us verify that the vector given in (3.1.10) is indeed a solution to the AVI. First we need to check that $\lambda \in K$. The second half of the LCP-like problem gives us the relation

$$\begin{aligned} s_{\mathscr{I}} &= H_{\mathscr{I}} H_{\mathscr{A}}^{-1} s_{\mathscr{A}} + H_{\mathscr{I}} H_{\mathscr{A}}^{-1} b_{\mathscr{A}} - b_{\mathscr{I}} \\ &= H_{\mathscr{I}}\lambda - b_{\mathscr{I}}. \end{aligned}$$

Combining this with (3.1.10) gives us

$$H\lambda = s + b. \tag{3.1.11}$$

Since by definition $s \geq 0$, this ensures that $\lambda \in K$. Now let us check that the generalized equation (3.1.1) holds with λ given by (3.1.10). First remember that the solution of (3.1.9) gives us s the slack variable from (3.1.11), and also r such that $-H^T r$ is an element of the normal cone $N_K(\lambda)$. The first half of the LCP-like problem gives us the relation

$$\begin{aligned} r_{\mathscr{A}} &= -H_{\mathscr{A}}^{-T} H_{\mathscr{I}}^T r_{\mathscr{I}} + H_{\mathscr{A}}^{-T} M H_{\mathscr{A}}^{-1} s_{\mathscr{A}} + H_{\mathscr{A}}^{-T} M H_{\mathscr{A}}^{-1} b_{\mathscr{A}} + H_{\mathscr{A}}^{-T} q \\ H_{\mathscr{A}}^T r_{\mathscr{A}} + H_{\mathscr{I}}^T r_{\mathscr{I}} &= M H_{\mathscr{A}}^{-1} s_{\mathscr{A}} + M H_{\mathscr{A}}^{-1} b_{\mathscr{A}} + q \end{aligned}$$

$$H^T r = M\lambda + q$$
$$0 = M\lambda + q - H^T r.$$

We have now checked that there exists en element $(-H^T r)$ of the normal cone of K at λ such that the generalized equation (3.1.1) holds. Hence, the vector given in (3.1.10) is a solution to the AVI.

Moving on to the algorithmic part, note that when μ goes to 0, a solution to (3.1.9) is at hand, as with Lemke's algorithm. Also if $\bar{q} \geq 0$, the problem is already solved: $w = \bar{q}$ and $z = 0$. If μ is positive, the idea is to take one of variable r_i or s_i out of w by setting it to zero and we exchange it with another one from the vector z, which then become non-negative. Hence we always have $w \geq 0$ and $z = 0$, which guarantees that the complementarity constraint $0 \leq w \perp z \geq 0$ is always satisfied. When we swap variables between w and z, M and q have to be updated.

In the following, we suppose that $\bar{q} \not\geq 0$: whence, μ becomes non-zero, and one of the variables in w has to become 0. More precisely, it will be r_j, with $j \in \mathscr{A}$ chosen as $j = \arg\min\left\{-\bar{q}_i \mid \bar{q}_i < 0,\ 1 \leq i \leq p\right\}$. Both $\overline{M}$ and $\bar{q}$ have to be updated as in Step 2 below, except that the column $\overline{M}_{\bullet\beta}$ has to be replaced by d. Then we set β to be the position of s_j in z. We can now describe one iteration of the algorithm to solve this LCP-like problem.

Step 1: Find $\alpha = \arg\min_i \left\{-\dfrac{\bar{q}_i}{\overline{m}_{i\beta}} \mid \overline{m}_{i\beta} < 0\right\}$. This determines the variable that is going to be set to 0, which is said to be the blocking variable.

Step 2: Perform the update of M, q, also known as the pivoting step:

$$\text{with } i \neq \alpha,\ j \neq \beta \qquad \overline{m}_{ij} \leftarrow \overline{m}_{ij} - \frac{\overline{m}_{i\beta}}{\overline{m}_{\alpha\beta}}\overline{m}_{\alpha j}$$
$$\text{with } \quad j \neq \beta \qquad \overline{m}_{\alpha j} \leftarrow -\frac{\overline{m}_{\alpha j}}{\overline{m}_{\alpha\beta}}$$
$$\text{with } i \neq \alpha \qquad q_i \leftarrow q_i - \frac{\overline{m}_{i\beta}}{\overline{m}_{\alpha\beta}}q_\alpha \qquad \overline{m}_{i\beta} \leftarrow \frac{\overline{m}_{i\beta}}{\overline{m}_{\alpha\beta}}$$
$$q_\alpha \leftarrow -\frac{\bar{q}_\alpha}{\overline{m}_{\alpha\beta}} \qquad \overline{m}_{\alpha\beta} \leftarrow \frac{1}{\overline{m}_{\alpha\beta}}$$

Step 3: If the variable at w_α is μ, we are done since $\mu = 0$ now and this subproblem is solved with $w = \bar{q}$. Otherwise we have to update β or in other words select the variable which is going to be set to a non-negative value. That is, if we have r_j at

w_α, the variable that should leave z and enter w_α is s_j and vice-versa. This keeps the complementarity constraint $0 \leq r \perp s \geq 0$. Hence β is set to the position of the variable leaving z at the next iteration. Note that this is where this algorithm differs from Lemke's one. In the latter we would just set $\beta = \alpha$. And we go back to Step 1.

Once a solution to (3.1.9) has been found, we can recover a solution the AVI as

$$\lambda = H_{\mathscr{A}}^{-1}(s_{\mathscr{A}} + b_{\mathscr{A}}).$$

Let us state few remarks on the implementation of this algorithm. First the fact that if $\bar{q} \geq 0$ in (3.1.8) then we directly have a solution, indicates that the choice of the vertex v can greatly reduce the computational load, for instance by using a warm start. Here, a warm start or hot start means that we use the previous solution of the AVI or one of the closest vertex as v. Remember that we solve an AVI each time we want to compute the control input and note that as long as the system is not in the discrete-time sliding phase (as given in Definition (2.2.4)), two successive values of the control input may have some components equal. On the other hand if we always use the same vertex as v, most of the computations required to build (3.1.8) can be done offline. In some cases, like for linear systems, the matrix M of the AVI does not depend on the current time instant, only q does. Thus the only term that depends on the AVI data is $H_{\mathscr{A}}^{-T} q$. Therefore, if this computation has to be done in realtime, it may be interesting to consider this option.

An interesting special case is when the set K is box-shaped, a particular choice of v greatly reduces the computational load: K can be described as

$$\left\{ x \in \mathbb{R}^p \mid \begin{pmatrix} I \\ -I \end{pmatrix} x \geq \begin{pmatrix} lb \\ -ub \end{pmatrix} \right\}.$$

We can set $H_{\mathscr{A}} = I$, $H_{\mathscr{J}} = -I$, $b_{\mathscr{A}} = lb$ and $b_{\mathscr{J}} = ub$. Then we can write (3.1.8) as

$$\overline{M} = \begin{pmatrix} I & M \\ 0 & -I \end{pmatrix} \quad \text{and} \quad \bar{q} = \begin{pmatrix} M\, lb - q \\ -lb + ub \end{pmatrix}.$$

The solution is also easily recovered as $\lambda = s_{\mathscr{A}} + lb$. This is close to the decomposition in [89, Sections 6.3 and 8.7] and [20] that we already mentioned in Section 2.1.3. However it is worth noting that with latter method, a standard LCP is solved and The complementarity variables from the latter do not have the same interpretation as r and s. For instance, remember that the variable $\lambda = \mathrm{Sgn}(x)$ is split in positive λ^+ and negative part λ^-.

More details on the numerical aspects and solvers for this kind of problems can be found in [36, 27] and [3].

3.2 Control Input Computation for Nonlinear Systems

In Chapter 2, we mostly dealt with linear systems since we can exactly integrate these using the ZOH method. However, in most applications the dynamics is nonlinear and using the explicit Euler discretization fails to properly capture it. As mentioned in Section 1.4, the discretization scheme provides us with a recurrence relation

$$x_{k+1} = F(x_k, x_{k+1}, t_k, t_{k+1}, u_k), \tag{3.2.1}$$

which influences the quality of the control input, especially when the system states are close to the sliding surface. Suppose that this is the case: at time t_k, the control input is computed to bring the sliding variable to 0. If the approximated dynamics given by (3.2.1) is not accurate enough, the system states at time t_{k+1} will be far from the manifold. Some chattering might be seen on the control input, if the system states cross the sliding manifold during each sampling period. To alleviate those issues, we now present a discretization method named the $\vartheta - \gamma$ scheme, which is analogous to the trapezoidal rule, a well known quadrature method. Let us first state the type of systems we consider:

$$\dot{x} = f(x,t) + g(x,\lambda,t) \qquad f\colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n \tag{3.2.2}$$

$$\sigma = h(x) \qquad \text{with} \quad g\colon \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}^n$$

$$-\lambda \in \partial \mathfrak{h}_{-S}(\sigma) \text{ or } -\sigma \in \mathcal{N}_S(\lambda) \qquad h\colon \mathbb{R}^n \to \mathbb{R}^p, \tag{3.2.3}$$

where $S \subset \mathbb{R}^p$ is a bounded convex polytope where λ take values. The control input u is defined as a function of λ: most of the time, we have $u = \lambda$. However, it is possible for u and λ to have different sizes: remember the twisting algorithm from Section 2.3, where $u = G(\lambda_1 + \beta\lambda_2)$. Note that the discretization of the relation (3.2.2) is independent of the discretization of the inclusion (3.2.3), for which we shall only consider an implicit discretization.

3.2.1 Presentation of the $\vartheta - \gamma$ Scheme

In the following we assume that the control input (and therefore λ) is constant on every time interval $[t_k, t_{k+1})$. We also suppose that the solution to the ODE (3.2.2) is an absolutely continuous function. This enables us to write the following recurrence relation:

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(x,t) + g(x,\lambda_{k+1},t)\mathrm{d}t. \tag{3.2.4}$$

Now, finding a good recurrence relation boils down to deriving a good approximation to the integral. To avoid too heavy notations, we choose not to index the internal variables of the scheme with k and we let $\tau := t_{k+1} - t_k$. The standard $\vartheta - \gamma$ scheme is close to the trapezoidal rule, giving rise to the following approximations:

$$\int_{t_k}^{t_{k+1}} f(x,t)\mathrm{d}t \approx \tau\vartheta f(x_{k+1}, t_{k+1}) + \tau(1-\vartheta)f(x_k, t_k)$$
$$\int_{t_k}^{t_{k+1}} g(x,\lambda_{k+1},t)\mathrm{d}t \approx \tau\gamma g(x_{k+1},\lambda_{k+1},t_{k+1}) + \tau(1-\gamma)g(x_k,\lambda_{k+1},t_k).$$

Now combining the relation (3.2.4) with the two previous ones, we end up with the nonlinear system of equation:

$$\mathcal{R}_{\curlyvee}(x,\lambda) = 0 \tag{3.2.5}$$
$$\mathcal{R}_{\curlyvee}(x,\lambda) := x - x_k - \tau\vartheta f(x,t_{k+1}) - \tau(1-\vartheta)f(x_k,t_k) - \tau\gamma g(x,\lambda,t_{k+1}) - \tau(1-\gamma)g(x_k,\lambda,t_k),$$

with x and λ as unknowns and $\mathcal{R}_{\curlyvee}$ can be sought as a residual. In the sequel, we opt to tackle (3.2.5) as a root-finding problem and we choose to use a method based on the Newton-Raphson algorithm. It is built on the idea of solving a series of linear systems, which generates a sequence $\{x^\alpha, \lambda^\alpha\}_{\alpha\in\mathbb{N}}$ of successive iterates, converging to a root of the nonlinear function under suitable assumptions. The two sequences are initialized as follows:

$$x^0 = x_k \quad \text{and} \quad \lambda^0 = \lambda_k,$$

with λ_k the value of λ during the previous time interval $[t_{k-1}, t_k)$. In our context, those linear systems are based on a linearization of the nonlinear equation (3.2.5). Therefore we define the linearized version of the residual at $(x^\alpha, \lambda^\alpha)$ as

$$\mathcal{R}_{\curlyvee}^{\,\alpha}(x,\lambda) := \mathcal{R}_{\curlyvee}(x^\alpha,\lambda^\alpha) + [\nabla_x \mathcal{R}_{\curlyvee}(x^\alpha,\lambda^\alpha)](x-x^\alpha) + [\nabla_\lambda \mathcal{R}_{\curlyvee}(x^\alpha,\lambda^\alpha)](\lambda-\lambda^\alpha). \tag{3.2.6}$$

Let us introduce some matrices:

$$A^\alpha := \nabla_x f(x^\alpha,\lambda^\alpha) \qquad B^\alpha := \nabla_\lambda g(x^\alpha,\lambda^\alpha)$$
$$K^\alpha := \nabla_x g(x^\alpha,\lambda^\alpha) \qquad B_c^\alpha := \nabla_\lambda g(x_k,\lambda^\alpha),$$

which are related to the Jacobians of $\mathcal{R}_{\curlyvee}$ by the relations

$$J_x^\alpha := \nabla_x \mathcal{R}_{\curlyvee}(x^\alpha,\lambda^\alpha) = I - \tau\vartheta A^\alpha - \tau\gamma K^\alpha \tag{3.2.7}$$
$$J_\lambda^\alpha := \nabla_\lambda \mathcal{R}_{\curlyvee}(x^\alpha,\lambda^\alpha) = -\tau\gamma B^\alpha - (1-\gamma)\tau B_c^\alpha. \tag{3.2.8}$$

We now detail one iteration of the algorithm. We look for $x^{\alpha+1}$ and $\lambda^{\alpha+1}$ solutions to the linear system $\mathcal{R}_L^{\alpha}(x,\lambda) = 0$. Starting from (3.2.6) and using the notations from (3.2.7) and (3.2.8), we have

$$J_x^{\alpha}(x^{\alpha+1} - x^{\alpha}) = -\mathcal{R}_{\diagup\!\!\diagdown}(x^{\alpha},\lambda^{\alpha}) - J_{\lambda}^{\alpha}(\lambda^{\alpha+1} - \lambda^{\alpha}).$$

We suppose that the square matrix J_x^{α} is invertible, an assumption that holds for τ or ϑ and γ small enough, see (3.2.7). Then $x^{\alpha+1}$ can be expressed as the solution of the previous linear system:

$$x^{\alpha+1} = x^{\alpha} - (J_x^{\alpha})^{-1}\mathcal{R}_{\diagup\!\!\diagdown}(x^{\alpha},\lambda^{\alpha}) + (J_x^{\alpha})^{-1}J_{\lambda}^{\alpha}(\lambda^{\alpha+1} - \lambda^{\alpha}),$$

where $\lambda^{\alpha+1}$ is the only unknown in the right-hand side. Now with this relation between $x^{\alpha+1}$ and $\lambda^{\alpha+1}$, we can eliminate $x^{\alpha+1}$ and the normal cone inclusion (3.2.3) gives the VI

$$0 \in h(x^{\alpha+1}(\lambda)) + N_S(\lambda), \tag{3.2.9}$$

which can be solved using a dedicated solver. Yet we continue to detail the iteration of our algorithm by transforming this VI into an AVI. This is motivated by the fact that in many applications the sliding variable σ is a linear function of the state, in which case the VI (3.2.9) is already an AVI. On the other hand, one of the most successful approaches to solving such VI is to apply a Newton method known as Newton-Josephy, which, roughly speaking, is based on solving a sequence of AVI. To this end, let us linearize h at current iterate x^{α}:

$$h_L^{\alpha}(x) = h(x^{\alpha}) + C^{\alpha}(x - x^{\alpha}) \quad \text{with} \quad C^{\alpha} := \nabla_x h(x^{\alpha}).$$

Substituting this in (3.2.9) gives the AVI:

$$0 \in q_{\mathrm{AVI}}^{\alpha} + M_{\mathrm{AVI}}^{\alpha}\lambda + N_S(\lambda),$$

with

$$\begin{aligned} M_{\mathrm{AVI}}^{\alpha} &:= C^{\alpha}(J_x^{\alpha})^{-1}J_{\lambda}^{\alpha} \\ q_{\mathrm{AVI}}^{\alpha} &:= h(x^{\alpha}) - C^{\alpha}(J_x^{\alpha})^{-1}\mathcal{R}_{\diagup\!\!\diagdown}(x^{\alpha},\lambda^{\alpha}) + M_{\mathrm{AVI}}^{\alpha}\lambda^{\alpha}. \end{aligned}$$

The description of one iteration of the algorithm is now finished. The two stopping criteria are

$$\begin{aligned} |\mathcal{R}_{\diagup\!\!\diagdown}(x^{\alpha+1},\lambda^{\alpha+1}) - \mathcal{R}_L^{\alpha}(x^{\alpha+1},\lambda^{\alpha+1})| &\le \varepsilon \\ |h(x^{\alpha+1}) - h_L^{\alpha}(x^{\alpha+1})| &\le \varepsilon, \end{aligned}$$

for a given "tolerance" $\varepsilon > 0$. The second condition could be replaced by checking whether $\lambda^{\alpha+1}$ solves the (nonlinear) VI (3.2.9). We do not present a theoretical analysis for this algorithm. The two main steps would be the analysis of the Newton-Josephy method if the sliding variable is a nonlinear function of the state. This one can be found for instance in [36, Section 7.3]. The analysis of the root-finding method could follow the one used for the Newton-Raphson one.

3.3 The SICONOS platform

3.3.1 Overview of the SICONOS platform

SICONOS aims at providing a general and common tool for the simulation of nonsmooth problems in various scientific fields like Applied Mathematics, Mechanics, Control Theory, Electrical circuits, Robotics, ... It originated from the European project SICONOS IST–2001–37172, which lasted from 2002 to 2006. Most of the development has been done at INRIA Grenoble Rhône-Alpes by BIPOP team members and members of SED, a service dedicated to support software developments.

On Figure 3.1 we can see the layered architecture of the SICONOS platform. We give a

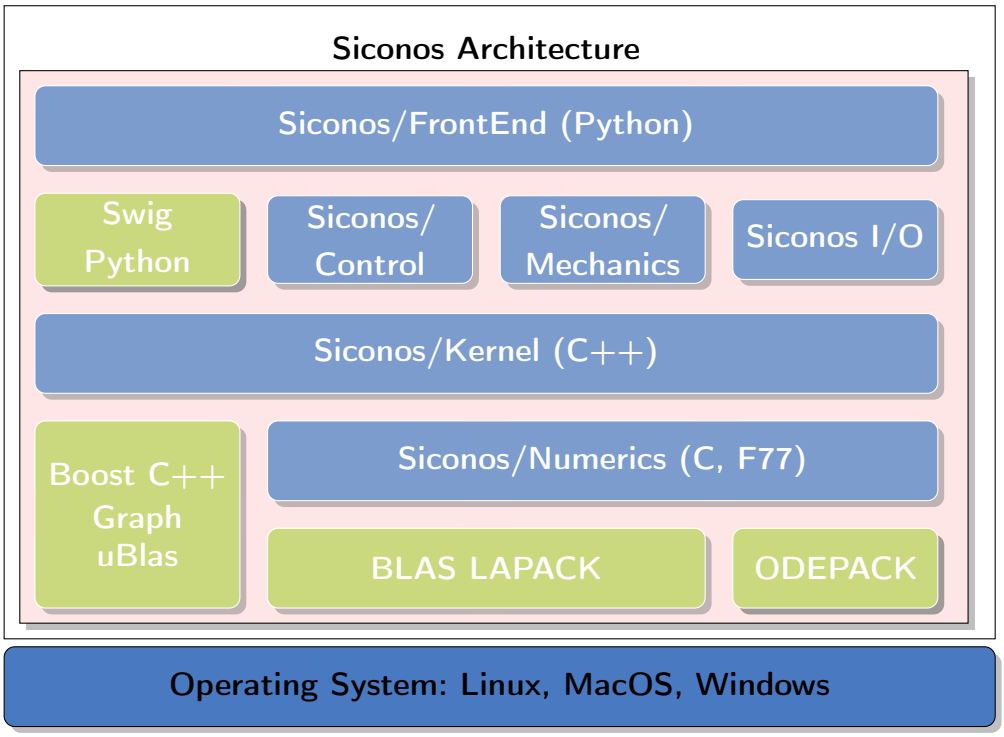


Figure 3.1: Synopsis of the SICONOS libraries

brief overview of the different modules as well as the work done by the author.

NUMERICS The optimization algorithms and numerical integration routines (ODEPACK, "Hairer's library") are the foundation of SICONOS. There are algorithms (implemented in C) for the following nonsmooth problems: LCP, MLCP, NCP, AVI, VI and friction contact problems. The main contribution of the author was the implementation of the algorithm for solving AVI described in Section 3.1.

KERNEL It is composed of three main parts: one deals with the *modeling* of dynamical systems, the second one is for the *simulation* of nonsmooth systems and the third one provides the data structure and also linear algebra routines. For the modeling part, we support 3 types of formalisms: the classical first order systems with $\dot{x} = f(x, t)$, the Lagrangian mechanics with generalized coordinates q and velocities $\dot{q}$ and the Newton-Euler formalism for multi-body systems.

To simulate a given nonsmooth dynamical system, two strategies are available: the event tracking and the event capturing methods. Some differences between those two types of schemes are given in Table 3.1. The first type of integrators are wrappers around existing code (LSODAR from ODEPACK[1] [56], HEM5[2] [16]) and when an event (like an impact) is detected, a nonsmooth problem (like an LCP) is solved. The second category has for representative the time-stepping algorithm by Moreau [87] which is implemented in C++, alongside other integrators. SICONOS can be used as a C++ library in which case the user specifies the simulation data and type of integrator in a C++ program. This module uses Numerics to solve the nonsmooth problems that were given during the modeling phase. The author implemented an integrator for linear first-order system based on the ZOH discretization and reworked the way events are processed.

CONTROL This module contains all control-related functionalities. A prototype was already available in the Kernel with just the base classes implemented. It has now SMC controllers and observers. We detail this module in Section 3.3.2.

MECHANICS This module is using the Newton-Euler formalism from the Kernel and provide some features like contact detection (for instance via Bullet) that are useful for multi-body simulations. It also contains specialized dynamical systems like disk, spheres and also objects to simulate various interactions (sphere-sphere, sphere-plan, ...).

[1] https://computation.llnl.gov/casc/odepack
[2] http://www.unige.ch/~hairer/software.html

I/O This module enables the user to save a given simulation in a file while the simulation is running and to resume it from the saved data. It provides also a visualization tool.

FRONTEND Python bindings are provided for all the aforementioned modules. Those are very convenient since they empower the user with the ease-of-use of the Python language and the whole scientific ecosystem. For instance the author was able to easily plot the simulation results with the Matplotlib library and to perform simulations with parameters varying in a given range. All the simulation results presented in this thesis are done in this way.

Let us focus on the event-driven and time-stepping simulation strategies. The first one is better known in the control community: it has been used to simulate hybrid systems where the events are defined when the system exists the domain of the current mode. Some of the weaknesses of this scheme like the inability to simulate system with Zeno phenomenon is well-known. On the other hand, a time-stepping scheme does not require such event-handling procedure and is able to "go through Zeno": such method can be proved to be convergent even in the presence of events accumulations. This kind of approach is perfectly suited for the computation of the control law when the latter is a step function: the fact that the control input value is constant during the sampling period is already part of our modeling. The pro and cons of the two strategies are summarized in the following table. To expand a bit on the time-stepping approach, it consists (roughly

Method	Advantages / Weaknesses
Event tracking schemes	⊕ high accuracy integration of event-free periods
(a.k.a event-driven)	⊖ no proof of convergence
	⊖ sensibility to numerical thresholds
	⊖ reformulation of constraints at higher degree
Event capturing schemes	⊕ robust, stable and proof of convergence (Zeno)
(a.k.a time-stepping)	⊕ able to deal with finite accumulation
	⊕ low kinematic level for the constraints
	⊖ low order of accuracy even in free flight

Table 3.1: Qualitative comparisons of time-stepping schemes for nonsmooth dynamics

speaking) in the time-discretization of the whole dynamical system: ODE and nonsmooth relation. While doing so, we form a One-Step NonSmooth Problem (OSNSP) which has to be solved at each time step. Hence the main stages of the process are:

1. integrate the dynamics without constraints, to get some "free" solutions.
2. construct and solve an OSNSP (an AVI for instance in SMC).
3. update the dynamics with the OSNSP solutions to get the full state update.

We detail in Section 3.2 this procedure in the context of the computation of the control input value.

3.3.2 Control Architecture in SICONOS

We now go into the specific of the simulation for control purposes with the canonical structure given in Figure 3.2.

We suppose that the plant we want to control is a continuous-time process and the con-

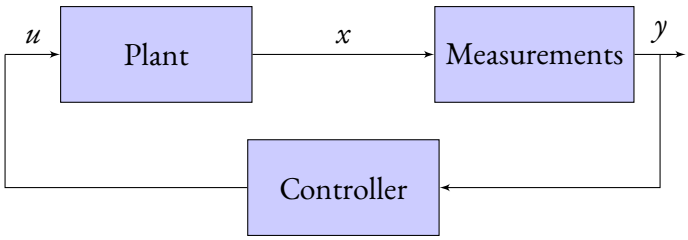


Figure 3.2: Classical control structure

troller (and observer) is implemented using a microprocessor. Hence we have two different "time domains": the plant, actuator and sensor are in continuous-time whereas the observer and controller are discrete-time. It is of paramount importance to keep this separation in the simulation: failure to do so seriously jeopardize the faithfulness of the simulation, for instance in the presence of nonlinear dynamics. It is also interesting to decouple the computation of the control input from the integration of the system to avoid performing the same approximations in both cases and hiding their effects. Just a note on the terminology: by *timestep* we refer to the difference between two time instants at which the solution to an ODE is evaluated. It is not always required to specify this timestep: suppose we want the solution of an ODE at time T starting from t_0 and we have little interest in the solution values inside this interval. Then we can use a smart integration routine like LSODAR or CVODE[3] [25], we need only to specify the interval $[t_0, T]$: the timestep, as well as the integration method, is chosen internally in order to provide an accurate result. In contrast the sampling period is fixed and indicates when the control input changes.

[3] https://computation.llnl.gov/casc/sundials/main.html

We took the following approach in the design of the Control toolbox: there are two types of simulations that run "simultaneously": first we use a high-precision integrator like LSODAR for the nonlinear continuous-time processes and the ZOH scheme for linear ones. In both cases, the contribution to the control input is added either to the derivative in the first case, or directly to the solution in the second one. The control input value has already been computed by the controller at the beginning of each sampling period, by the mean of a simulation with a nonsmooth dynamical system. This simulation has for timestep the sampling period while for the first one, the timestep is not required to be small: in the ZOH case we can perfectly integrate the system and an integration routine like LSODAR we in fact specify an interval over which the system has to be integrated. This interval only specifies the frequency at which we get values for the system states. We now go into some implementation details on functionalities implemented or enhanced by the author.

Managing the different parts of the simulation

The need to run concurrently at least two types of simulation schemes and to properly separate the different components in Figure 3.2 requires us to create an infrastructure to properly manage the run of a simulation. It consists in `Events` associated with the different parts of the simulation, which form a stack managed by the `EventsManager`. The concept of `Event` was already in SICONOS since with an event-driven scheme, the impacts are modeled as events. We add the following types of `Event`: `Sensor` for measurement-related actions, `Controller` for updating the control input and `Observer` to provide a new estimate of the state. Each `Event` has for attribute a time instant and a given priority, which enables us to provide a strict total order between the different types. Let `e1` and `e2` be two `Events`. They are compared in the following way:

`e1` $<$ `e2` if `e1.time` $<$ `e2.time` or if `e1.prio` $>$ `e2.prio` when `e1.time` $=$ `e2.time`

The comparison on the time instants is not done with floating-point arithmetic: we convert those to integers when we store this information. This enables us to form a stack of events: the time of the first `Event` defines the current time in the simulation. This `Event` has already been processed and is kept to give this reference in time. The second `Event` defines the action that is executed. First the integration of the dynamical system(s) is performed to the time of the `Event`. Then the action associated with the `Event` is performed: for a `TimeDiscretisation` it saves the current state, for a `Sensor` it gets the current state of the plant connected to it. With an `Observer` an update to the estimated state is performed

and with a `Controller` the control input is updated. Then the first `Event` gets updated: the timestep or sampling period is added to its current time and if the result is smaller or equal to the end time of the object, it is reintroduced in the stack.

There are a few hurdles to properly get the simulation running in this way. Firstly the update of the time of each `Event` has to be done with great care. The most natural way to proceed would be to add the sampling period or timestep to the current value. However, this operation is done thousand of times and the IEEE-754 standard used for floating point computation cannot guarantee that this operation is done exactly. And by reusing the previous result over and over, the errors accumulate and some drifting appear. Let us illustrate this by the following `C` program.

```c
#include <float.h>
#include <stdio.h>

int main(void)
{
  double h[] = {1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10, 1e-11, 1e-12};
  double t[12] = {0.};
  unsigned long long iter = 10;

  printf("timestep\t\tbad summation\t\t\t\tbetter summation:\n");
  for(unsigned i = 0; i < 12; ++i)
  {
    for(unsigned long long j = 0; j < iter; ++j)
    {
      t[i] += h[i];
    }
    double tmul = 0.0 + h[i]*iter;
    printf("h = %e\tt =  %.*e\tt = %.*e\n", h[i], DECIMAL_DIG, t[i], DECIMAL_DIG, tmul);
    iter *= 10;
  }
  return 0;
}
```

The output is given in Table 3.2. The increasing drift is striking: the value of time should

timestep	bad summation	better summation:
h = 1.000000e-01	t = 9.99999999999998889777e-01	t = 1.00000000000000000000e+00
h = 1.000000e-02	t = 1.00000000000000666134e+00	t = 1.00000000000000000000e+00
h = 1.000000e-03	t = 1.00000000000000666134e+00	t = 1.00000000000000000000e+00
h = 1.000000e-04	t = 9.99999999999061861544e-01	t = 1.00000000000000000000e+00
h = 1.000000e-05	t = 9.99999999980837550595e-01	t = 1.00000000000000000000e+00
h = 1.000000e-06	t = 1.00000000007918110612e+00	t = 1.00000000000000000000e+00
h = 1.000000e-07	t = 9.99999997501699544600e-01	t = 1.00000000000000000000e+00
h = 1.000000e-08	t = 1.00000000289867184757e+00	t = 1.00000000000000000000e+00
h = 1.000000e-09	t = 9.99999925399328803977e-01	t = 1.00000000000000000000e+00
h = 1.000000e-10	t = 1.00000006947562347647 8e+00	t = 1.00000000000000000000e+00
h = 1.000000e-11	t = 1.00000008206489465756 3e+00	t = 9.99999999999998889777e-01
h = 1.000000e-12	t = 9.99984408876058106230 3e-01	t = 1.00000000000000000000e+00

Table 3.2: Drift occuring with different summations strategies

always be 1, as it is almost always the case with the better summation strategy. Furthermore it is important to note that this drift can go in both direction, depending on the added number. In the case where the sampling period is 1ms for the control-related `Event` and the timestep of the simulation is 0.1ms, the drifts are in opposite directions. This induces a lack of precision but also generates issues within the integration process. An integration routine like LSODAR terminates with a fatal error if asked to integrate over an interval with a small length. When this drift occurs, we will end up in this situation. The proper solution was to make the update in the following way: given a period Δ, we compute time instants as $t_k = t_0 + k\Delta$. This eliminates the drift but does not rule out another floating-point issue: it is possible that the for a given time instant multiple of both the sampling period and the timestep, the floating-point numbers do not coincide. This error does not propagate: at the next time instant the two `Events` should coincide, they will. However this can induce a fatal error: for instance with LSODAR, we will ask to integrate between the two `Events` and the routine will refuse and terminates with an error. To prevent such case, we set the same time instant for two `Events` if they differ by less than a given tolerance.

Implementation of the Control toolbox

For the control-related functionalities, we implemented an integrator using the ZOH scheme. For an LTI system, the update of the state variable is given by

$$x_{k+1} = A^* x_k + B^* u_k,$$

where A^* and B^* are computed as

$$A^* := \exp(Ah) \quad \text{and} \quad B^* := \int_0^h \exp(A(h-\tau))B\mathrm{d}\tau,$$

with A and B the matrices from the classical representation. Following the guidelines in [86], those two matrices are computed as the solution $X \in \mathbb{R}^{n\times n}$ and $Y \in \mathbb{R}^{n\times n}$ evaluated at time h to the matrix ODEs

$$\begin{aligned} \dot{X} &= AX \quad \text{and} \quad \dot{Y} = AY + B \\ X(0) &= I \quad \text{and} \quad Y(0) = 0. \end{aligned}$$

LSODAR is the integration scheme used to get those matrices and the solutions are computed one column at a time. Given the way these matrices are computed, it is only interesting to compute them in the LTI case with all `Events` synchronized, that is there is

no delay between them. Even if only the sampling period varies or if the solution x has to be evaluated between two time instants, it is better to directly integrate the dynamics using LSODAR instead or computing first A^* and B^* and then update the state value.

The second integrator we use to run the simulation is based on the LSODAR integration routine. This one is more versatile than the previous one, since we can evaluate the solution x at any time. This enables us to model delays for instance on the Sensor or the Controller and to consider also nonlinear dynamics.

For the computation of the control input value, the `EulerMoreauOSI` integrator is used. It is based on the $\vartheta - \gamma$ scheme we presented earlier. The only addition here was to be able to simulate a nonlinear system with a contribution of the control of the form $g(x)u$ or $g(x, u)$, that is with a state-dependent function g.

Those integrators belong to the `Kernel` module and may be used for in very different contexts. Hence, we do not want to put this control-specific part in the integrators. The solution was to add an interface for additional contributions to the dynamics. By default, there is no action, but we can in the Control Simulation classes, we set one according to the type of integrator. The contributions fall into 2 main categories: we either add a term to the derivative $\dot{x}$ like for LSODAR or to the solution x_{k+1} as for ZOH.

To finish this part on the programming side, let us digress a bit on the object-oriented programming (OOP) paradigm. In order to be able to simulate the fully nonlinear case, that is nonlinear dynamics and a sliding variable defined as a nonlinear function of the state, an overhaul of the way computations on the nonsmooth relation are done was required. The reason was that in the Newton loop of this algorithm, we need to perform computation using the iterates x^α and λ^α. However with OOP, the values are usually stored as attribute of the object, and the computation of a quantity tends to have side effects like the update of some internal quantity, which are undesirable in our case since that would erase for instance x_k, the current value of the state. Some parts of SICONOS have been reworked to move away from this paradigm. A practically functional style[4] was used instead, that is the functions used to compute quantities operates only on their arguments. This removes the problem of unwanted side effects like the update of a class attribute while computing an iterate of the Newton loop.

[4]See for instance John Carmack's post on http://gamasutra.com/view/news/169296/Indepth_Functional_programming_in_C.php

3.4 Numerical Analysis of the Control Input Discretization

In this section we present a thorough numerical analysis of the influence on the closed-loop behavior with different types of discretization of both u^{eq} and u^s. We also illustrate some results from Section 2.2, for the order of convergence of the different discretization methods. We consider a 2D system, in order to plot the state evolution, and which enables us to show a variety of behaviors. We ended up with this system:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) & A = \begin{pmatrix} 0 & 1 \\ 19 & -2 \end{pmatrix}, \\ \sigma = Cx & \\ u(t) = u^{eq}(t) + u^s(t) & B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ C^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{cases} \tag{3.4.1}$$

The matrix A has the eigenvalues $\lambda_1 = 3.47$ and $\lambda_2 = -5.47$. The dynamics on the sliding surface is given by $\Pi A = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}$, which has eigenvalues 0 and -1. Through this section, we chose $\alpha = 1$. The initial state is $(-15, 20)^T$. The first set of simulations uses a sampling period 0.3 s for the control and the second one a sampling period 0.03 s. The simulations run for 150 s and were carried out with the open source SICONOS software package [2][5], as described in the previous section. Figures were created using Matplotlib [62].

Let us provide simulation results not only for the controller studied in Section 2.2 but also for some other inputs. The objective in this section is to provide an overview of the different behaviors of the closed-loop system when various discretization methods are used. A more formal study of their properties and performances is done in Section 2.2.3. From all the possible time-discretization schemes, we focus on the one-step explicit, implicit, and midpoint ones. With the expressions for u^{eq} and u^s in (1.1.2) and (1.1.3), the proposed discretized values for the equivalent control u^{eq}_k are:

$$u^{eq}_{k,e} = -(CB)^{-1}CAx_k \qquad \text{explicit input,} \tag{3.4.2a}$$

$$u^{eq}_{k,i} = -(CB)^{-1}CAx_{k+1} \qquad \text{implicit input,} \tag{3.4.2b}$$

$$u^{eq}_{k,m} = 1/2(u^{eq}_{k,e} + u^{eq}_{k,i}) \qquad \text{midpoint input,} \tag{3.4.2c}$$

and the two possibilities for the discontinuous control u^s_k are:

$$-u^s_k = \alpha \operatorname{sgn}(\sigma_k) \qquad \text{explicit input,} \tag{3.4.3a}$$

$$-u^s_k \in \alpha \operatorname{Sgn}(\sigma_{k+1}) \qquad \text{implicit input.} \tag{3.4.3b}$$

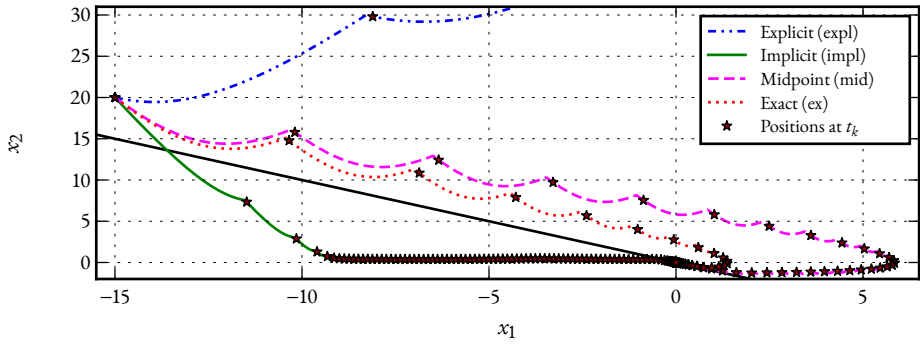
[5] http://siconos.gforge.inria.fr

We use the single-valued signum function in (3.4.3a) since the case $\sigma_k = 0$ is not worth considering for explicit inputs. Moreover with the set-valued Sgn function, if $\sigma_k = 0$, then we would have $\mathrm{Sgn}(\sigma_k) \in [-\alpha, \alpha]^p$ and there is no proper selection procedure to get a value for u_k^s, whereas the selection procedure in the implicit case is presented in Section 2.2. The most commonly used control law is the combination of (3.4.2a) and (3.4.3a). This kind of discretization has been studied in [45, 47, 114], with a focus on the sequence formed by σ_k once the system state approaches the sliding manifold. The value of (3.4.2b) is detailed in Section 2.2.3. The ZOH sampled-data version of the system (3.4.1) is used for the discrete-time dynamics. In Section 3.4.1, the nominal system (3.4.1) is simulated and in a matching perturbation is added in Section 3.4.2. For each set of simulations, three types of figures are shown. The first one presents an overview of the trajectories of the different closed-loop systems (like Figures 3.1 and 3.4). The next one displays some details around the origin (Figures 3.2, 3.5 and 3.7). Finally, we present plots of the different discontinuous inputs (Figure 3.3, 3.6 and 3.8). Markers are also added to help visualize the position of the closed-loop system at some of the time instants t_k.

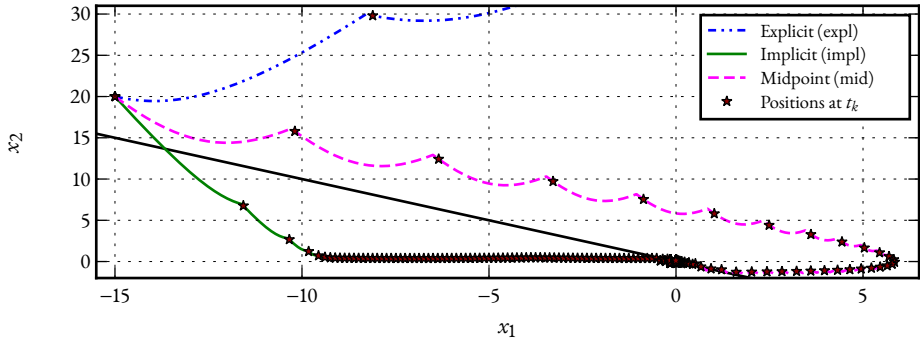
3.4.1 Nominal case

The trajectories for the different closed-loop systems are plotted on Figure 3.1. The motion in the reaching phase depends only on the discretization method used for the equivalent control u^{eq}. It is only near the sliding manifold that the discretization method of the discontinuous control u^s plays a role. If the explicit scheme in (3.4.2a) is used for the discretization of u^{eq}, the system diverges (Figure 3.1a and 3.1b, curves (ei) and (ee)). This discretization method can destabilize a system which is stable in continuous time. If the implicit scheme in (3.4.2b) is used for the discretization of u^{eq}, then the discretization error may not affect stability but it can induce some unexpected behavior. As we can see in Figure 3.1, curves (ii) and (ie), the trajectories are crossing the sliding manifold. This phenomenon can be explained by the following fact: let Δ_k be the discretization error on u^{eq} at time t_k. We have the relation $\sigma_{k+1} = \sigma_k + \Delta_k + CB^* u_k^s$. Let us consider the implicit discretization of u^s. If $0 < \sigma_k < CB^*$, then the system should enter the discrete-time sliding phase. However if $\Delta_k + \sigma_k < -2CB^*$, then for any value of u_k^s, $\sigma_{k+1} < -CB^*$. Hence, due to the discretization error, u^s fails to bring σ_{k+1} to 0 and the trajectory of the system crosses the sliding manifold. The same happens with the explicit discretization of u^s. With the midpoint method in (3.4.2c), curves (mi) and (me), and with the new control scheme (2.2.13), curve (ex), the system state reaches the sliding manifold directly.

Near the sliding manifold (Figures 3.2a and 3.2b), the behavior of the system is more



(a) Implicit discretization of u^s. (ei) is for pair (3.4.2a)–(3.4.3b); (ii) for (3.4.2b)–(3.4.3b); (mi) for (3.4.2c)–(3.4.3b); (ex) for (2.2.13).

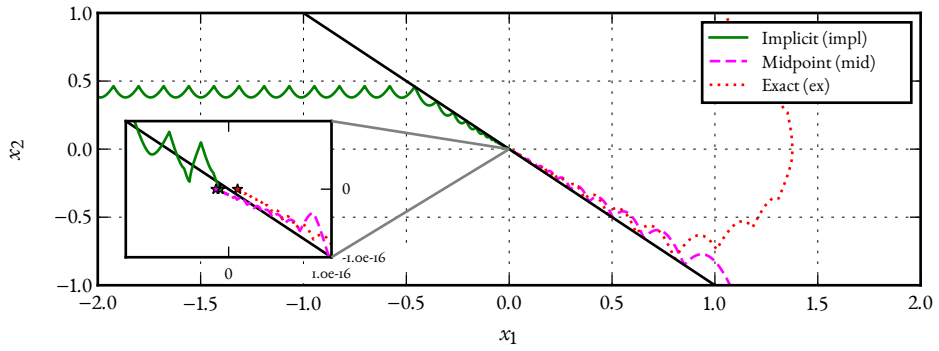


(b) Explicit discretization of u^s. (ee) is for pair (3.4.2a)–(3.4.3a); (ie) for (3.4.2b)–(3.4.3a); (me) for (3.4.2c)–(3.4.3a).

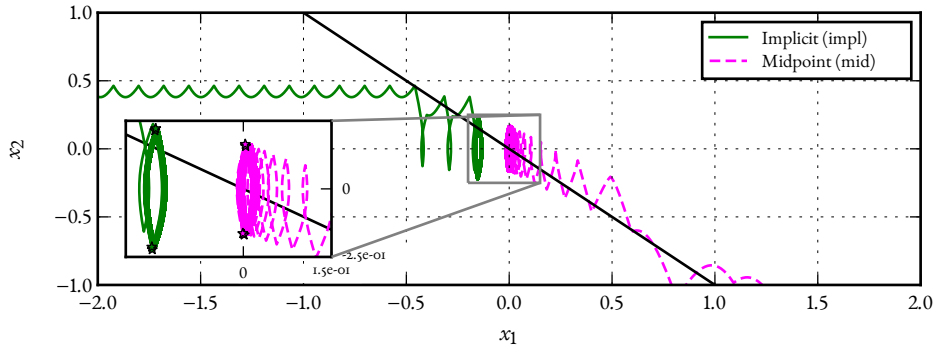
Figure 3.1: Simulations of system (3.4.1) using different discretization methods, with $h = 0.3$ s and $\alpha = 1$.

sensitive to the discretization of u^s. In the implicit case (method (3.4.3b), Figure 3.2a) and in the discrete-time sliding phase, σ_k is very close to 0 ($\sigma_k = 0$ with the exact method). In each case, the state converges to the origin (at the machine precision). This is visible on the zoom box in Figure 3.2a, where markers indicate the state of the system at each time instant t_k, during the last second of each simulation. When the explicit method (3.4.3a) is used, the system chatters around the sliding manifold, within a neighborhood of order h (0.3 s here), see Figure 3.2b.

In Figure 3.3b, the explicitly discretized discontinuous control u^s takes its values in $\{-1, 1\}$ and starts at some point a limit cycle, as studied in [47]. This cycle is also visible on the zoom box in Figure 3.2b with the help of the markers. In Figure 3.3a, for each discretization of u^{eq}, u^s converges to 0, which is the value that u^s_{cont} takes in the sliding phase. In the implicit and midpoint cases, at the beginning of the discrete-time sliding phase, u^s takes non zero values since there are discretization errors on u^{eq}. That is, if $\sigma_k = 0$, $\sigma_{k+1} \neq 0$. The discontinuous control tries to bring σ_{k+1} to 0 and counteracts the effects of

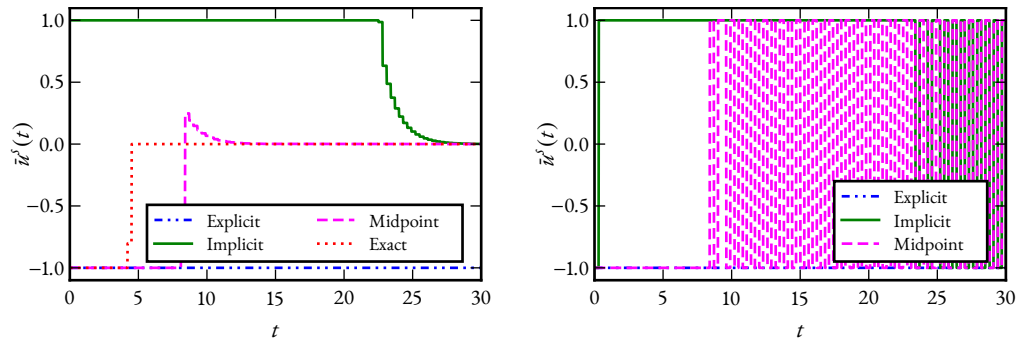


(a) Implicit discretization of u^s



(b) Explicit discretization of u^s

Figure 3.2: Detail of Figure 3.1, $h = 0.3$ s, $\alpha = 1$.



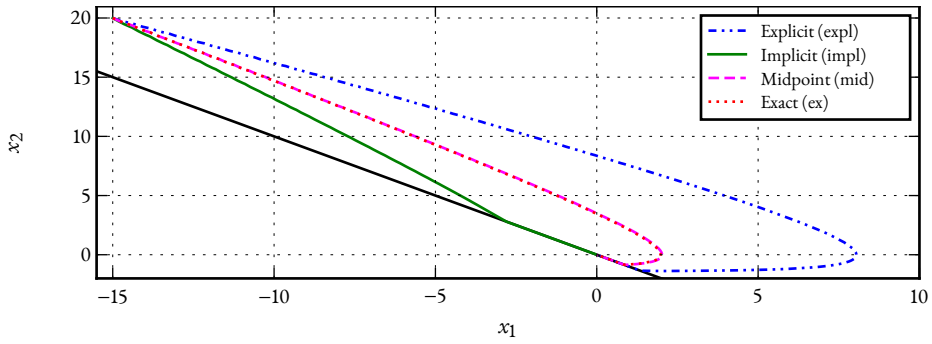
(a) Implicit discretization of u^s

(b) Explicit discretization of u^s

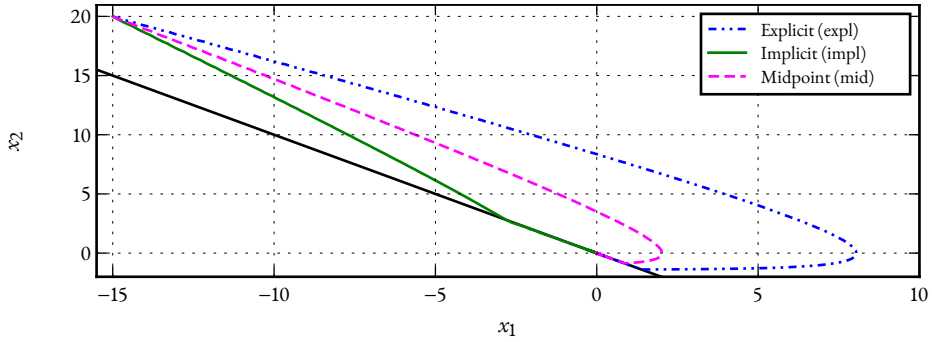
Figure 3.3: Evolution of u^s for different discretization methods, with $h = 0.3$ s and $\alpha = 1$.

the error. As the state goes to the origin, the error converges to 0. The simulation results seem to indicate that the discretization error is smaller in the midpoint case than in the implicit case. This observation is formally stated in Lemma 2.2.17. With the exact method of Section 2.2.1, u^s goes to 0 after one sampling period in the discrete-time sliding phase. In Figures 3.3a and 3.3b, with the explicit discretization of u^{eq}, u^s takes always the same value, since the closed-loop system moves away from the sliding manifold. In terms of

convergence to the sliding manifold, the first closed-loop system to enter the discrete-time sliding phase is the exact method (Figure 3.3a), then the midpoint, finally the implicit method. With the explicit method on u^{eq}, the system moves away from the sliding manifold and thus cannot enter the discrete-time sliding phase.



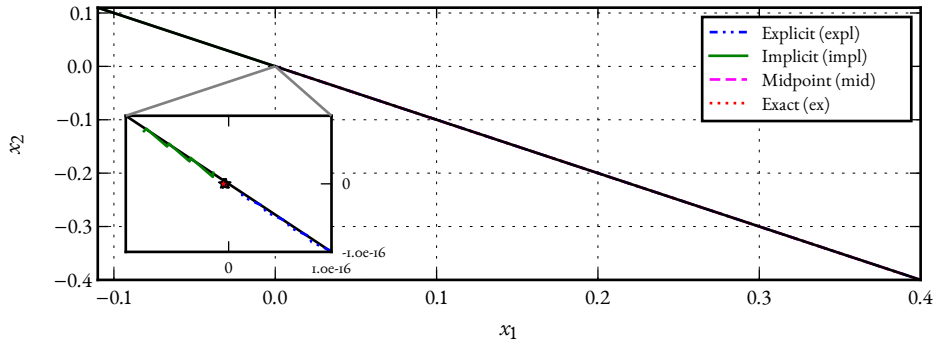
(a) Implicit discretization of u^s



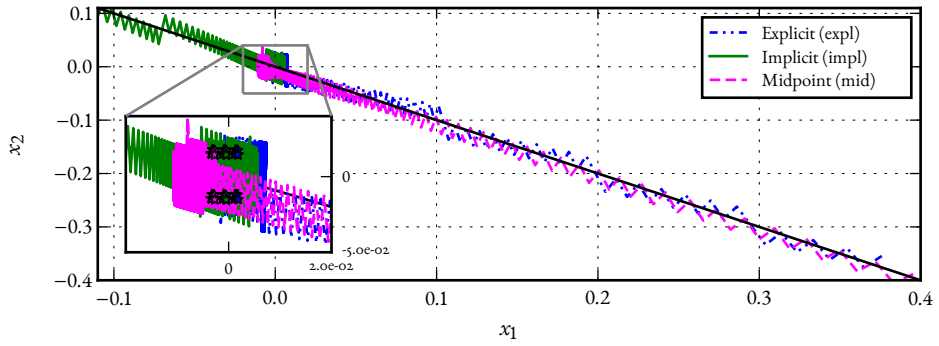
(b) Explicit discretization of u^s

Figure 3.4: Simulations of system (3.4.1) using different discretization methods, with $h = 0.03$ s and $\alpha = 1$.

The next set of simulations uses the same parameters as the previous one, except for the sampling period which is smaller: $h = 0.03$ s. In contrast with the results presented in Figure 3.1, the closed-loop system is stable in all cases, see Figure 3.4. As expected, the discretization error is smaller and no trajectory crosses the sliding manifold. It is not possible to distinguish the solution associated with the midpoint method from the one obtained with the exact method in Figure 3.4a. In Figure 3.5a with the implicit discretization of u^s, the states converge again to a very small ball near the origin. In the explicit case, there is some numerical chattering, again with the same order of magnitude as the sampling period ($h = 0.03$ s, Figure 3.5b). In Figure 3.6a, once in the discrete-time sliding phase, u^s counteracts the discretization error on u^{eq}, which is smaller than in Figure 3.3a. The discretization error for the midpoint discretization in (3.4.2c) is much smaller, and its curve

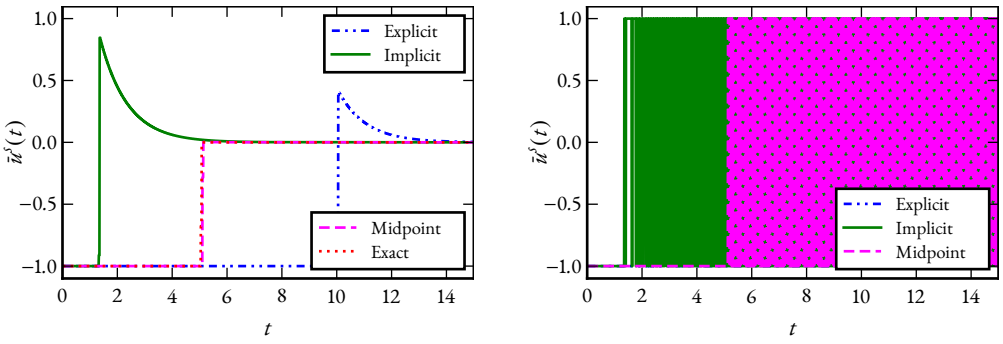


(a) Implicit discretization of u^s



(b) Explicit discretization of u^s

Figure 3.5: Detail of Figure 3.4, $h = 0.03$ s, $\alpha = 1$.



(a) Implicit discretization of u^s

(b) Explicit discretization of u^s

Figure 3.6: Evolution of u^s for different discretization methods, with $h = 0.03$ s and $\alpha = 1$.

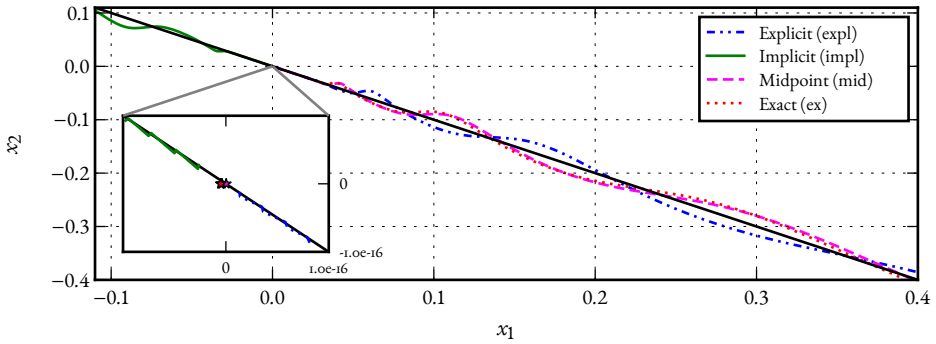
overlaps completely with the one of the exact discretization method. In Figure 3.6b the same bang-bang behavior as in Figure 3.3b is seen, with a higher frequency of switching.

The results presented here bring into view the numerical chattering caused by an explicit discretization of u^s, while the implicit method is free of it. The importance of the discretization of u^{eq} is also illustrated, with the explicit method leading to a diverging system and the counterintuitive behavior yielded by the implicit method. An analysis of the

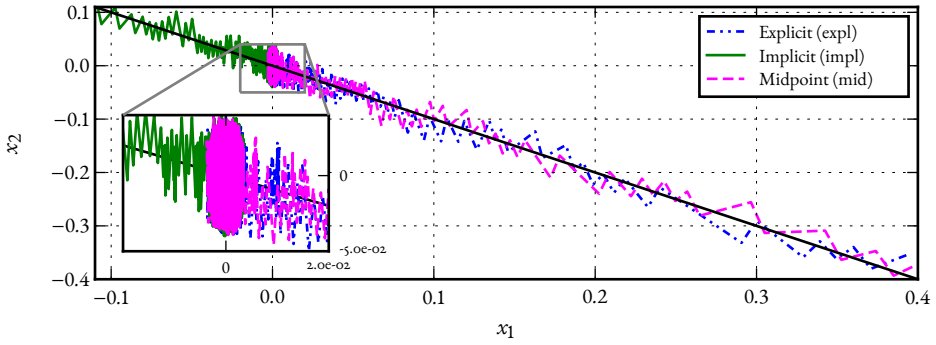
different phenomena is provided in the next section. The exact method from Section 2.2.1 produces good results and in agreement with the theoretical results.

3.4.2 Perturbed case

We now add a perturbation in the system (3.4.1), which takes the form $\xi(t) = 0.6\exp(\min(6-t, 0))\sin(2\pi t)$ in the next set of simulations. Note that for all t, $|\xi(t)| \leq 0.6$. This particular ξ has been chosen to highlight that if the perturbation vanishes, with the implicit discretization in (3.4.3b), u^s goes also to 0, whereas in the explicit case (3.4.3a), u^s continues to switch between -1 and 1. With the implicit discretization of u^s (Figure 3.7a) the



(a) Implicit discretization of u^s

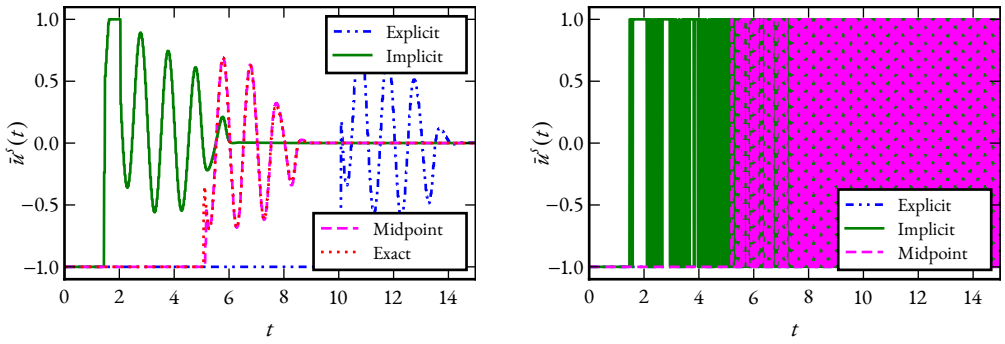


(b) Explicit discretization of u^s

Figure 3.7: Simulations of system (3.4.1) using different discretization methods for u^{eq} and $h = 0.03$ s (perturbed case).

closed-loop system enters the discrete-time sliding phase at some point. Recall that in this case, if the assumptions in Proposition 2.2.11 are satisfied, then $u_k^s = -(CB^*)^{-1}Cp_{k-1}$. It takes such value in order to counteract the effect of the perturbation during the elapsed time interval, hence imitating the solutions defined using Filippov's framework. However the trajectories are now clearly only in a neighborhood of the sliding manifold. Finally in each case in Figure 3.8a, u_k^s settles to 0, as in continuous time. Indeed, the perturbation ξ

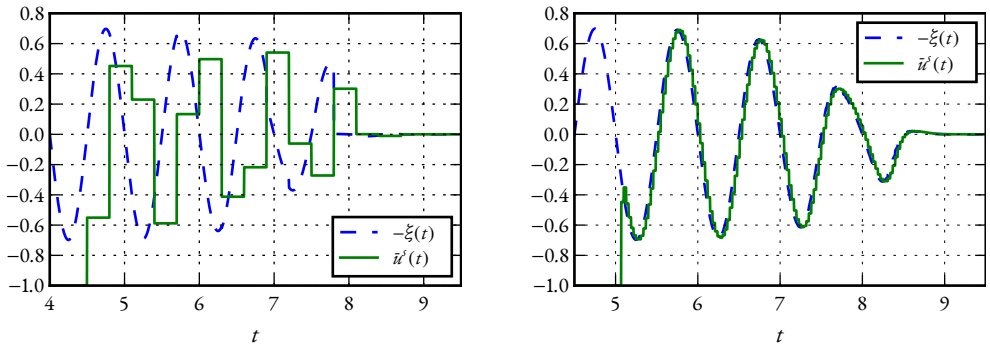
used in this simulation goes to 0 exponentially fast at some point. On the other hand, with an explicit discretization of u^s (Figure 3.8b), it is much harder to see the influence of the perturbation on u^s since filtering would be necessary to see the effect. The control input chattering is striking with the explicit discretization in Figures 3.3b, 3.6b, and 3.8b. We shall see the same difference in the analysis of the experimental results, like for instance on Figures 4.5, 4.3 and 4.2. In Figure 3.9 we further illustrate the phenomenon in the implicit



(a) Implicit discretization of u^s

(b) Explicit discretization of u^s

Figure 3.8: Evolution of u^s for different discretization methods for u^{eq} and u^s, $h = 0.03$ s. (perturbed case)



(a) $h = 0.3$ s

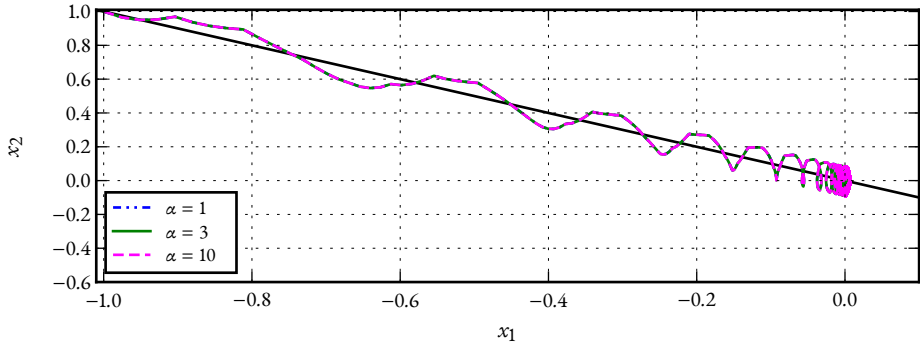
(b) $h = 0.03$ s

Figure 3.9: Evolution of u^s and the perturbation using the new control scheme for two different sampling periods.

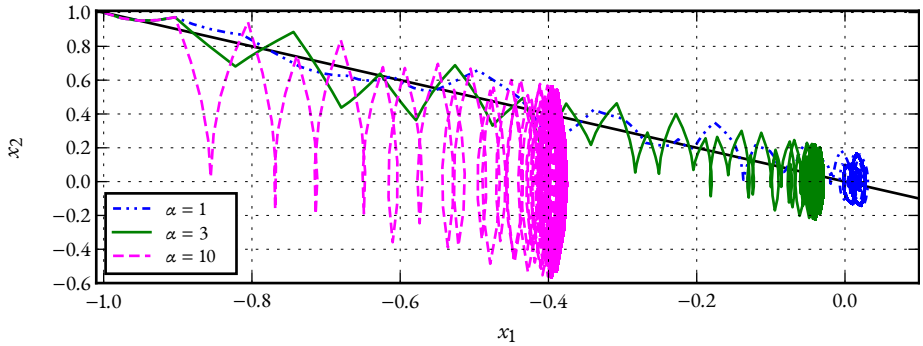
case: u^s approximates $-\xi$ with a delay proportional to h. This illustrates the convergence of u^s to u^s_{cont}, as stated in Proposition 2.2.15 in Section 2.2.2.

The simulation results displayed in Figure 3.10 and 3.11 illustrate that with an implicit discretization of u^s, the chattering in the discrete-time sliding phase is solely due to the perturbation. The setup is the same as in Section 3.4.1, except that there is a perturbation $\xi(t) = 0.9\sin(t)$ and α, the magnitude of the discontinuous control, changes. For the

present set of simulations, we use $\alpha = 1, 3$ and 10, values large enough to ensure that the action of the perturbation is always dominated by the control. On Figure 3.10a, we cannot distinguish the three trajectories in the discrete-time sliding phase, since even if u_k^δ takes value in a larger set, the selected value within (α, α) does not change. This is supported by the Figure 3.11a: it is again not possible to differentiate the values taken by the controllers. On the contrary, in Figure 3.10b, three different trajectories are clearly visible. Here we conclude that the numerical chattering is dominant: each time we increase α, the amplitudes of the oscillation around the manifold are getting bigger. The precision of the system seems to be affected by the magnitude of the control. The bigger it is, the farther the system oscillates from the origin.



(a) Implicit discretization of u^δ.

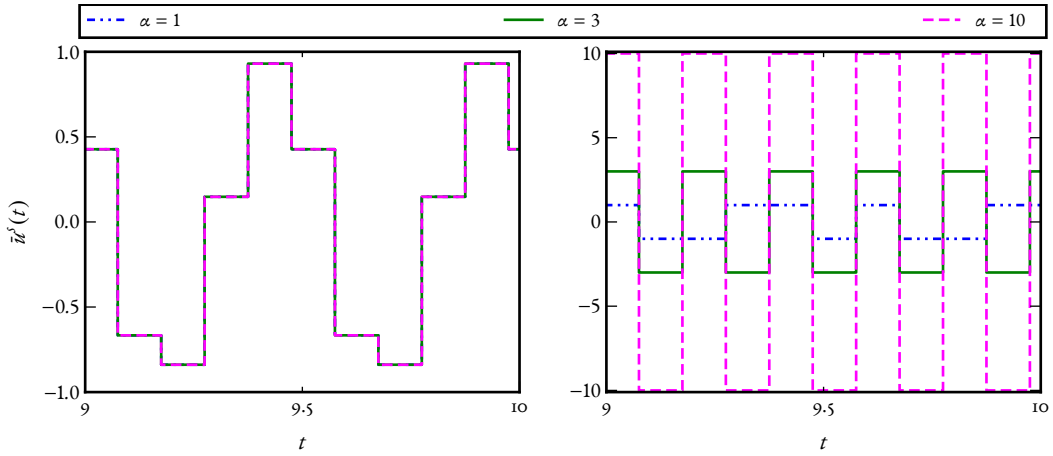


(b) Explicit discretization of u^δ.

Figure 3.10: Simulations of system (3.4.1) with a perturbation, using different values for α and with $h = 0.1$s.

3.4.3 Comparison with saturated SMC

Now we would like to compare the controller from Section 2.2.1 against an implementation using the "saturation trick", on a system subjected to perturbation. A proper numerical analysis of the saturated case is difficult. Therefore we settled for a numerical



(a) Implicit discretization of u^s.

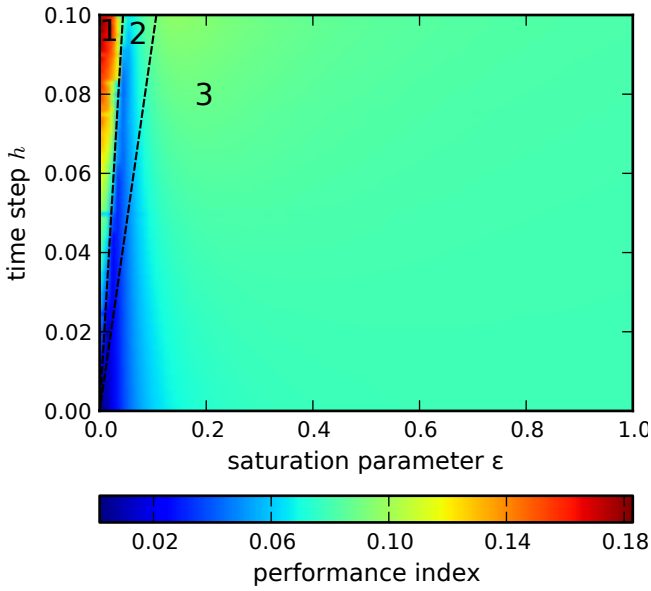
(b) Explicit discretization of u^s.

Figure 3.11: Evolution of u^s, using different values for α and with $h = 0.1$ s.

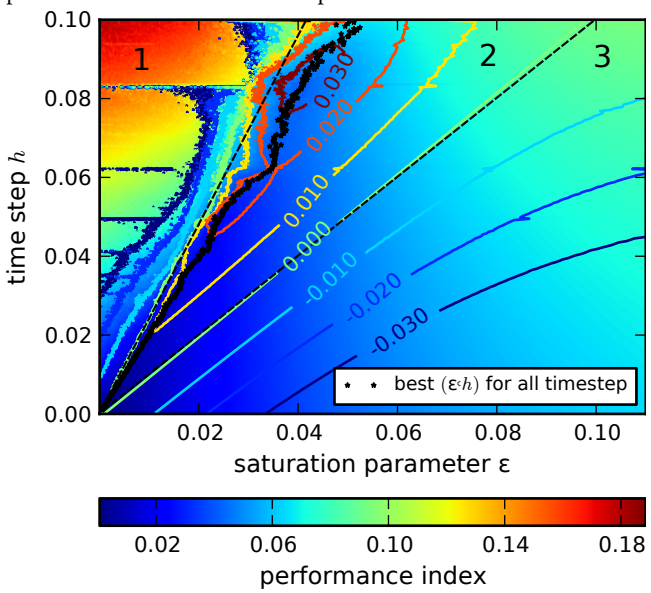
experiment: we simulate the system (3.4.1) with a perturbation $\xi(t) = \sin 4\pi t$. Instead of using a discontinuous control, we use the following input: $u^s_{\text{cont}}(t) = -\operatorname{sat}_\varepsilon(\sigma(t))$ with $\operatorname{sat}_\varepsilon(x) = \begin{cases} x/\varepsilon & \text{if } |x| \leq \varepsilon \\ \operatorname{sgn}(x) & \text{if } |x| > \varepsilon \end{cases}$. Each simulation lasts 150s. We use two metrics to measure the performance of the different controllers. To measure the (output) chattering, due to the discretization and the perturbation, we sum the absolute value of the sliding variable σ_k for the last 20s: $C_1 := \sum_k |\sigma_k|$. To measure the control effort (or input chattering), we measure the variation of the control for the last 20s: $C_2 := \operatorname{Var}^T_{T-20}(u^s)$. Each quantity defines the performance index in Figure 3.12 or 3.13. We choose to consider only the last 20s of each simulation to capture the behavior near the sliding manifold. Let us recall that without perturbation, the implicit controller always supersedes the saturated explicit one, since it suppresses numerical chattering and $u^s_k = 0$ in the discrete-time sliding phase.

With both indexes, we can divide the space into 3 cones, numbered 1, 2 and 3 in Figure 3.12 and 3.13. This separation helps us to compare both controllers. In Figure 3.12a the performance in terms of chattering is presented. For large values of ε, the chattering does not change when the sampling period varies: the control action does not attenuate the effect of the perturbation. With a small ε, the behavior is more complex, as depicted in Figure 3.12b. On Figure 3.12b, the overall best performance is obtained with small values for both ε and h. However for small values of ε, the performance can degrade rapidly if the sampling period h is not small enough, as seen in Region 1. The dark points indicate for each value of h the pair (ε, h) of parameters yielding the best performance. It seems that

there is a linear relationship between those values. However it is unclear if this observation on one particular system remains valid with a different perturbation. The level sets in Figure 3.12b are used to compare the performance of the implicit and the saturated explicit controllers. On Figure 3.13, the performance in terms of control cost is presented. The best performance is achieved for large ε since the slope of the saturated function is gentle. On the other hand in Figure 3.13b, with a small ε, the cost increases and explodes with ε close to 0, as in Region 1. The level sets indicate the difference between the costs of the two different controllers. It is worth noting that in Region 2 where the saturated controller is better in Figure 3.12b, it has a higher cost in term of control (Figure 3.13b). In Region 3, where the saturated controller performs less in terms of chattering (Figure 3.12a), it has a smaller cost in terms of control (Figure 3.13a). Indeed with a large ε, the control input is small when the closed-loop system is close to the sliding manifold. The cost is then very small, but the disturbance is not attenuated at all. The implicit controller appeals to us as the best compromise between the input and output chattering. It is also very easy to use, since it requires no particular tuning with respect to the sampling period or the perturbation.

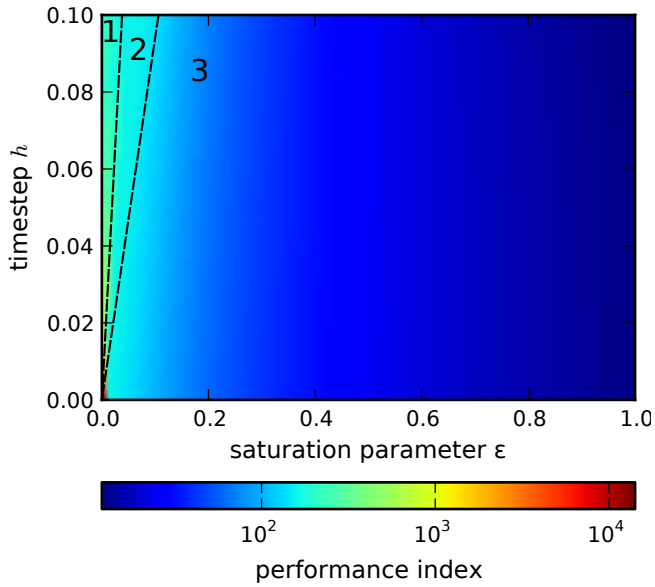


(a) Simulation results with 100 regularly spaced values for the sampling period h and 100 logarithmically spaced values for the saturation parameter ε.

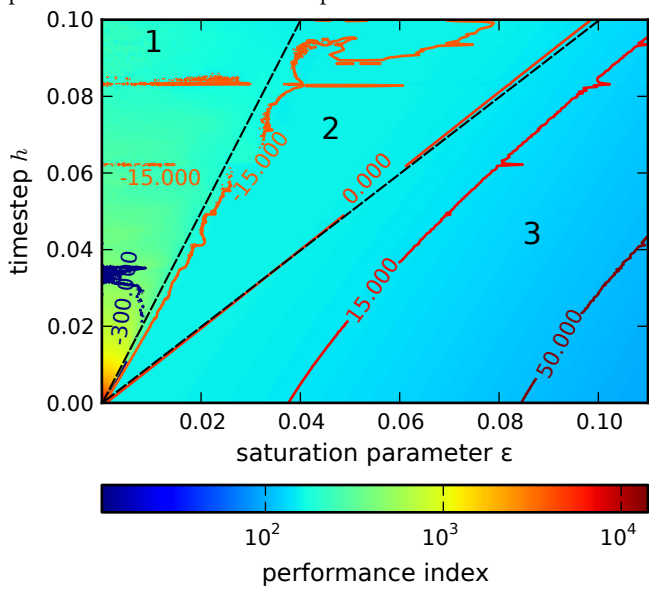


(b) Detail of Figure 3.12a, 300 values for h and 1000 values for ε, forming a regular grid. Level sets were also added to show the difference in performance between the implicit discretization and the explicit one with saturation. If the difference is positive, the explicit saturated control is performing better than the implicit one.

Figure 3.12: Simulation results of a perturbed system controlled using sliding mode with a saturation. The performance index is the sum of the $|\sigma_k|$ for the last 20s.



(a) Simulation results with 100 regularly spaced values for the sampling period h and 100 logarithmically spaced values for the saturation parameter ε.



(b) Detail of Figure 3.13a, 300 values for h and 1000 values for ε, forming a regular grid. Level sets were also added to show the difference in performance between the implicit discretization and the explicit one with saturation. If the difference is positive, the explicit saturated control is performing better than the implicit one.

Figure 3.13: Simulation results of the same perturbed system controlled using sliding mode with a saturation. The performance index is the sum of the $|u^s_{k+1} - u^s_k|$ for the last 20s.

3.5 Simulation of a Nonlinear System

Let us now present some simulation results on a nonlinear system to illustrate the ϑ–γ scheme presented in Section 3.2 while using the architecture exposed in Section 3.3. We choose to simulate the electropneumatic setup on which we implemented both the classical ECB-SMC and the twisting controller. A detailed presentation of the dynamics and the experimental results is available in the next chapter, in Section 4.1. Let us just start with the dynamics of a piston, with two chambers P and N and a mass attached to the end of the rod:

$$\begin{aligned} \dot{p}_P &= \frac{\varkappa r T}{V_P(y)}[\varphi_P + \psi_P u - \frac{S}{rT} p_P v] \\ \dot{p}_N &= \frac{\varkappa r T}{V_N(y)}[\varphi_N - \psi_N u + \frac{S}{rT} p_N v] \\ \dot{v} &= \frac{1}{M}[S\left(p_P - p_N\right) - b_v v] \\ \dot{y} &= v, \end{aligned} \tag{3.5.1}$$

with p_P (resp. p_N) two pressures, y and v being the position and velocity of a moving mass. The constant $\varkappa$ is the polytropic index, r the ideal gas constant, T the temperature (supposed the same inside and outside the chambers) and b_v the viscous friction coefficient. The volumes in each chamber are V_P and V_N, both depending on the actuator position y. The constant piston section is S. Finally, φ_X and ψ_X (X being P or N) are both 5^{th} order polynomial functions versus p_X.

Compared to the experimental setup, we consider that the whole state is known and that there is no perturbation. We want to illustrate the accuracy provided by the ϑ–γ method with respect to a crude approximation of the nonlinear dynamics. To this end, we shall compare three different controllers:

– An implicitly discretized twisting algorithm, in a "classical" way as in (2.3.3) with the ϑ–γ scheme. The control input is computed as

$$-\lambda_{k+1} \in \mathrm{Sgn}\begin{pmatrix}\sigma_{k+1}\\ \dot{\sigma}_{k+1}\end{pmatrix} \quad \text{and} \quad u_k = G(\lambda_{1,k+1} + \beta\lambda_{2,k+1})$$

– The same controller, but without the ϑ–γ scheme: to compute the control input at time t_k, the dynamics is linearized around x_k.

– The twisting algorithm as studied in Section 2.3.4 and defined by (2.3.26) and (2.3.28), coupled with the ϑ–γ scheme. The computation of the control input value is then

$$-\lambda \in \partial \mathfrak{h}_{-K}\begin{pmatrix}\sigma_{k+1}\\ \dot{\sigma}_{k+1}\end{pmatrix} \quad \text{and} \quad u = G(\lambda_1 + \beta\lambda_2),$$

$$\text{with} \quad K = \left\{x \in \mathbb{R}^2 \mid Ex \geq b\right\} \quad \text{and} \quad E = \begin{pmatrix}1 & 0\\ -h/2 & 1\\ -1 & 0\\ h/2 & -1\end{pmatrix}, b = \begin{pmatrix}-1\\ -1\\ -1\\ -1\end{pmatrix}.$$

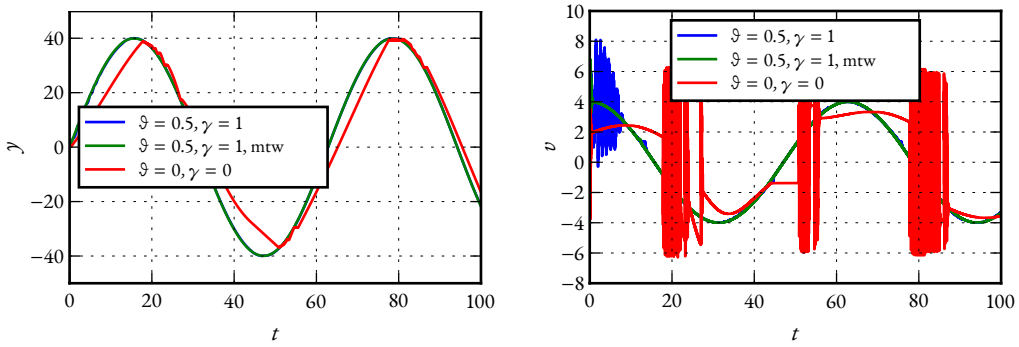
Note that the three controllers are defined in an implicit way. We do not consider an explicit discretization of the control input. The control objective is to make the mass at the end of the piston track a given trajectory, which is sinusoidal here. Therefore, we want to design the control input such as to set the tracking error $e := y - y_d$ to 0. From the dynamics in (3.5.1), it is easy to see that the relative degree is 3 between e and u. Thus we need to define our control input in the following way:

$$\sigma := \alpha e + \dot{e} \quad \text{and} \quad \dot{\sigma} = \alpha\dot{e} + \ddot{e}$$

Details on the control strategy for this setup can be found in Section 4.1.1. We just mention that in continuous-time, the parameter α influences the speed of convergence for the tracking error e: once the sliding phase $\sigma = 0$ is established, the evolution of the error is governed by the ODE $\alpha e + \dot{e} = 0$. Then increasing α speeds up the convergence of e towards 0.

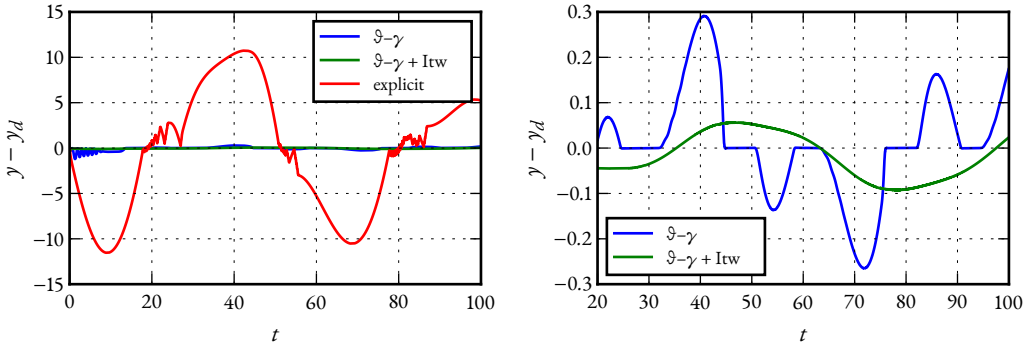
Let us now present some simulation results with the following parameters values: $\vartheta = 0.5, \gamma = 1$, $G = 10^{-2}$, $\alpha = 10$ and $h = 0.1s$. The sampling period is chosen deliberately large to highlight the good performances of the ϑ–γ scheme, even in such setup. The initial state is $y = v = 0$ and values of the pressures p_P and p_N are selected such that to ensure that we start from a (physical) equilibrium.

On Figure 3.1 the evolution of the position and the speed of the load is depicted. The control input computed with the explicit discretization of the dynamics performs poorly compared to the two others: on Figure 3.1a the tracking error is clearly larger than with the other two controllers. This controller fails to generate a sinusoidal trajectory, whereas the two others managed to do so. On Figure 3.1b, we can see some chattering appearing at some point with this controller, which is never the case with the two others. Let us continue with the tracking error on y as depicted on Figure 3.2: on the left, it is



(a) Evolution of the position (in mm). (b) Evolution of the speed (in mm/s).

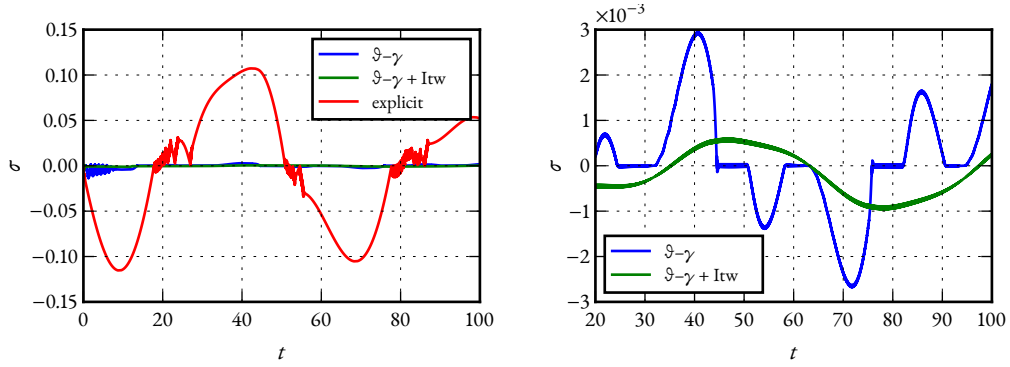
Figure 3.1: Simulations results for the system (3.5.1) with the 3 controllers.



(a) Tracking error $y - y_d$ (in mm). (b) Detail of the left figure.

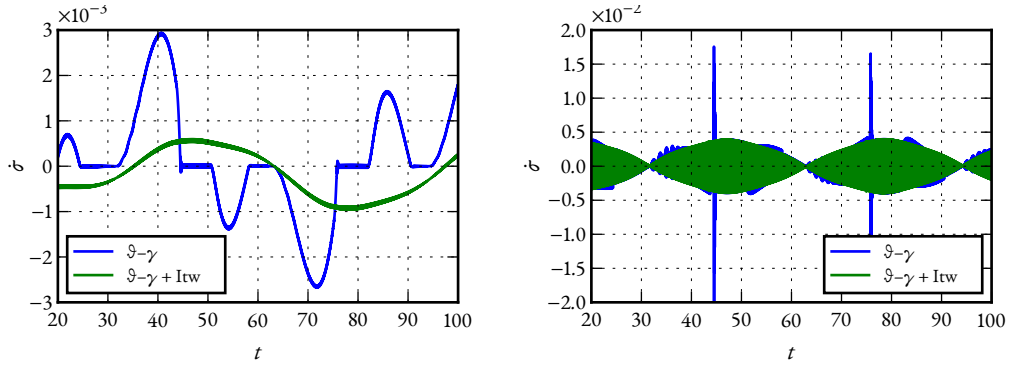
Figure 3.2: Tracking error on position, on the left with the 3 controllers, on the right only with the 2 using the ϑ–γ scheme.

easy to conclude that the tracking performance of the controller with explicitly discretized dynamics is an order of magnitude worst than the two others. On the right, Figure 3.2b, the controllers with the Newton loop are indeed performing better. We can see a difference between those two: the sliding variable with the modified one has a smaller magnitude than with the simply discretized controller. On the other hand the latter manages to keep the error to much smaller values during some period. Moving on to the sliding variables, on Figure 3.3 we again see the benefit of the ϑ–γ scheme. The controller using this method alongside the modified twisting algorithm is able to keep σ two orders of magnitude smaller than the controller with the basic discretization. On the right picture we see that two controllers yield a sliding variable σ with a shape similar to the tracking error. The evolution of the second sliding variable $\dot{\sigma}$ can be found in Figure 3.4. On the left, we can see again the chattering behavior with the explicitly discretized dynamics. Between the two other controllers, there is not much difference, except for some spikes. With Figure 3.5, we



(a) Sliding variable. (b) Detail of the left figure.

Figure 3.3: Sliding variable σ evolution: on the left with the 3 controllers, on the right only with the 2 using the ϑ–γ scheme.

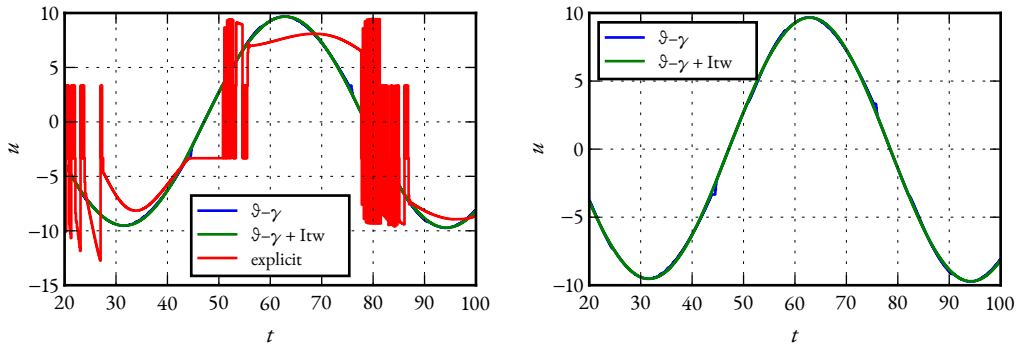


(a) Tracking error $y - y_d$ (in mm). (b) Detail of the left figure.

Figure 3.4: Evolution of the sliding variable $\dot{\sigma}$, on the left with the 3 controllers, on the right only with the 2 using the ϑ–γ scheme.

look at the control input values. The chattering phenomenon we have been noticing can also be seen here. The magnitude of the control input remains the same for all controllers. The chattering-like behavior is mostly due to the high-frequency switching of one control input.

To sum up, the following benefits from using the ϑ–γ scheme are illustrated: better tracking performances and no chattering-like behavior. Note that we are going to see those degraded performances again when investigating the experimental results for this system in Section 4.1. We shall see in Section 4.1.3 that some parameters, including α, have to be carefully tuned in order to get good results. With a better integration of the dynamics, it appears that the controller performance may be less sensitive to the parameter α. This could be interesting to investigate, as well as the differences between the two twisting controllers.



(a) Control input value (in mV).

(b) Detail of the left figure.

Figure 3.5: Control input evolution, on the left with the 3 controllers, on the right only with the 2 using the ϑ–γ scheme.

Conclusion

In this chapter we presented the numerical workhorse for the simulation and computation of the control input: First we gave algorithms to solve the AVI(K, q, M) whose solution is either the control input value or part of it, like in the twisting case. In particular we gave a detailed description of the algorithm by Cao and Ferris which finds a solution for every q and M whenever the set K is closed convex bounded. We also show with the ϑ–γ scheme how one could deal with nonlinear dynamics in a more efficient way than just discretizing it explicitly. We discussed some implementation details within the SICONOS platform, with a particular attention given to the Control toolbox that emerged as one of the outcomes of this PhD. Those software developments enabled us to realize the numerical study in Section 3.4, where we illustrate some theoretical results from Section 2.2. In the last section, we highlight the importance of the discretization of the nonlinear dynamics, which influences the quality of the control input given by the controller. With the large sampling period we considered, the advantages of using this method are nicely illustrated. We use this opportunity to put into action the modified discrete-time controller we proposed in Section 2.3. The results look promising and further studies should be undertaken as well as an implementation on the experimental setup. The next chapter is exactly devoted to the presentation of experimental results with both the ECB-SMC and the twisting controller. However we did not perform the implementation of neither the ϑ–γ scheme, nor the modified twisting controller since at the time the experiments had to be done, we lacked perspective on those two aspects and we opted to first implement the basic version. It turned out that it was already a challenge to properly tune the different parameters in the control loop.

Chapter 4

Experimental Results with Sliding Mode Controllers

This last chapter is dedicated to the experimental validations of the discrete-time sliding mode controllers presented in Chapter 2. We already illustrated the good behavior and some theoretical results in Chapter 3, but in Control it is important to go beyond simulation and to test the control laws on experimental setups. Those experiments were conducted as part of the ChaSlim project, founded by the ANR[1]. This project is a collaboration between the NON-A team at INRIA Lille – Nord Europe, the IRCCyN lab in Nantes and the BIPOP team at INRIA Grenoble Rhône-Alpes. We present experimental results for two setups: an electropneumatic actuator from the IRCCyN lab and an inverted pendulum at École Centrale de Lille. We tested both the classical sliding mode controller (but without an equivalent part) and the twisting algorithm.

In the presentation we focus on the discretization of the $\mathrm{Sgn}(\cdot)$ multifunction since it is this one that influences the most the behavior of the control loop. Therefore, we mainly compare the results obtained with an implicit discretization of the $\mathrm{Sgn}(\cdot)$ multifunction versus an explicit discretization. Both systems have nonlinear dynamics which could be discretized with the ϑ–γ scheme presented in Section 3.2. But we wanted first to get results on the discretization of controllers and collecting good data proved to be time-consuming. The fact that none of the setups is located at INRIA Grenoble did not help testing the control laws. This is why we only present results with the same discretization of the dynamics.

[1] ANR BLANC ANR–11–BS03–0007 http://chaslim.gforge.inria.fr/

4.1 Electropneumatic System

In this section, we present results from an implementation of both explicit and implicit twisting controllers on an electropneumatic plant, as depicted on Figure 4.1. This setup

Figure 4.1: Electropneumatic system

is located at the IRCCyN lab (École Centrale de Nantes, France). The control problem at hand is the tracking of a sinusoidal trajectory for the position of the end of the piston, and this despite the perturbation induced by the other piston. In the following, we first present the model of this system and the twisting control law that we use. Note that we used only the one obtained by the implicit discretization and not the one discussed in Section 2.3.4. The later was studied after the experiments were done. Then we move to the analysis of the captured data, highlighting the importance of the discretization process which is unfortunately often overlooked. We also discuss the influence of some parameters, like the choice of the sliding variable, on the closed-loop behaviour.

4.1.1 Setup Description

Plant Dynamics, Actuators and Sensors

Let us now present the physical system, actuators and sensors. The electropneumatic

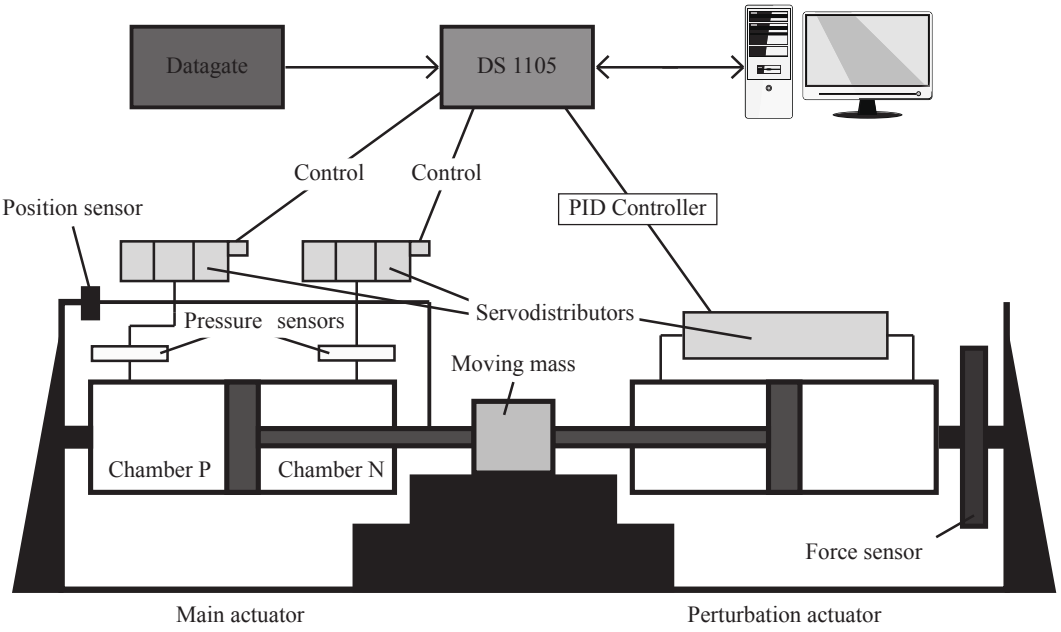


Figure 4.2: Schematic of the electropneumatic system

system, as depicted on Figure 4.2, has two actuators. On the left-hand side, there is a double acting electropneumatic actuator (the "main" one) controlled by two servodistributors and composed of two chambers denoted P and N. The piston diameter is 80 mm and the rod diameter is 25 mm. With a source pressure equal to 7 bars, the maximum force developed by the actuator is 2720 N. The air mass flow rates entering the chambers are modulated by two three-way servodistributors. The pneumatic jack horizontally moves a load carriage of mass M. This carriage is coupled with the second electropneumatic actuator, the "perturbation" one, on the right-hand side. The goal of the latter is to impress a dynamic load force on the main actuator. This actuator has the same mechanical characteristics as the main one, but the air mass flow rate is modulated by a single five-way servodistributor. The control variable u is constrained to take values between -10 and 10 volts. The position y, the pressures p_P, p_N are available but both the speed v and acceleration are computed using a filtered differentiator given in frequency domain by

$$D(s) = \frac{s}{1 + \tau s}. \tag{4.1.1}$$

Under some assumptions detailed in [100], the plant dynamics can be written as a nonlinear system affine in the control input $[u_P \; u_N]^T$, with u_P (resp. u_N) the control input of the servo distributor connected to the P (resp. N) chamber. The model is divided in two parts: the first two equations describe the pressure dynamics in each chamber and the motion of the piston is given by the last two equations. There is a single control

objective which is to force the load position to track a reference trajectory. Therefore we set $u \coloneqq u_P = -u_N$, and the dynamics of the electropneumatic experimental setup is

$$\begin{aligned} \dot{p}_P &= \frac{\varkappa r T}{V_P(y)}[\varphi_P + \psi_P u - \frac{S}{rT} p_P v] \\ \dot{p}_N &= \frac{\varkappa r T}{V_N(y)}[\varphi_N - \psi_N u + \frac{S}{rT} p_N v] \\ \dot{v} &= \frac{1}{M}[S\left(p_P - p_N\right) - b_v v - F] \\ \dot{y} &= v, \end{aligned} \tag{4.1.2}$$

with p_P (resp. p_N) the pressure in the P (resp. N) chamber, y and v being the position and velocity of the load. The constant $\varkappa$ is the polytropic index, r the ideal gas constant, T the temperature (supposed the same inside and outside the chambers) and b_v the viscous friction coefficient. The volumes in each chamber are V_P and V_N, both depending on the actuator position y. The constant piston section is S. The external force applied by the perturbation actuator is denoted by F. Finally, φ_X and ψ_X (X being P or N) are both 5^{th} order polynomial functions versus p_X [98], that characterize the mass flow rate q_X in the chamber X in the following way

$$q_X = \varphi_X(p_X) + \psi_X(p_X, \operatorname{sgn}(u_X))u_X.$$

The sources of uncertainty can be the polytropic index $\varkappa$, the mass flow, the temperature T, the mass M, the viscous friction coefficient b_v and the disturbance force F. They can be modeled by additive bounded functions added to the nominal part of each parameter. As an example, the mass M can be viewed as the sum of a nominal part and an uncertain one: $M =: M_n + \Delta M$, where ΔM is a bounded uncertainty and M_n the nominal value.

Control Strategy

The presence of uncertainties motivates the use of a sliding mode control scheme, well-known for its robustness. A first study was already conducted for equivalent-based sliding mode controller, with a comparison between explicit, implicit and saturation methods [112]. The experiments we present here were carried on with the discrete-time twisting controller presented in Section 1.1.2. Since we are interested in a tracking problem for the position of the load, y is the variable to be controlled. The desired position of the piston is y_d and the position error in the tracking problem is $e \coloneqq y - y_d$. The choice of this output leads to a relative degree 3. Therefore, to bring the relative degree between the sliding

variable and the control input to 2, so as to apply the twisting algorithm, we define the sliding variable as

$$\sigma := \alpha e + \dot{e}. \tag{4.1.3}$$

Its first and second derivatives are

$$\dot{\sigma} = \alpha\dot{e} + \ddot{e} \text{ and } \ddot{\sigma} = \alpha\ddot{e} + y^{(3)} - y_d^{(3)},$$

where

$$y^{(3)} = \ddot{v} = \frac{1}{\mathcal{M}}[S\left(\dot{p}_P - \dot{p}_N\right) - b_v\dot{v} - \dot{F}].$$

Using the relation in (4.1.2), we get

$$\begin{aligned} y^{(3)} = &\frac{S\varkappa rT}{\mathcal{M}}\left(\frac{\varphi_P}{V_P} - \frac{\varphi_N}{V_N}\right) - \frac{S^2\varkappa}{\mathcal{M}}\left(\frac{p_P}{V_P} + \frac{p_N}{V_N}\right)v \\ &- \frac{b_v}{\mathcal{M}^2}\left(S(p_P - p_N) - b_v v - F\right) - \frac{\dot{F}}{\mathcal{M}} \\ &+ \frac{S\varkappa rT}{\mathcal{M}}\left(\frac{\psi_P}{V_P} + \frac{\psi_N}{V_N}\right)u. \end{aligned} \tag{4.1.4}$$

Let us define the following functions

$$\begin{aligned} \Phi := &\frac{S\varkappa rT}{\mathcal{M}}\left(\frac{\varphi_P}{V_P} - \frac{\varphi_N}{V_N}\right) - \frac{S^2\varkappa}{\mathcal{M}}\left(\frac{p_P}{V_P} + \frac{p_N}{V_N}\right)v \\ &- \frac{b_v}{\mathcal{M}^2}\left(S(p_P - p_N) - b_v v\right) + \alpha\ddot{e} - y_d^{(3)} \end{aligned}$$

and

$$\Psi := \frac{S\varkappa rT}{\mathcal{M}}\left(\frac{\psi_P}{V_P} + \frac{\psi_N}{V_N}\right). \tag{4.1.5}$$

Finally, the sliding variable dynamics is

$$\ddot{\sigma} = \Phi + \Delta\Phi + (\Psi + \Delta\Psi)u \tag{4.1.6}$$

given that we consider that all the uncertainties are "additive", that is the vector fields can be written as the sum of a nominal part Φ and Ψ and uncertain terms $\Delta\Phi$ and $\Delta\Psi$. The latter include for instance the modeling errors and the action of the perturbation actuator like F and $\dot{F}$ in (4.1.4).

The implicit controller is constructed in the following way: the control input is discretized using the implicit discretization, that is

$$u_k = G(1\,\beta)\lambda \quad \text{and} \quad -\lambda \in \operatorname{Sgn}(\widetilde{\Sigma}_{k+1}), \tag{4.1.7}$$

with $\beta = 2/3$ and $\widetilde{\Sigma}_{k+1} = (\widetilde{\sigma}_{k+1}, \widetilde{\dot{\sigma}}_{k+1})$ the value of the sliding variables given by discrete-time dynamics that we now derive. Writing the sliding variable dynamics as a first-order ODE, we get

$$\dot{\Sigma} = A\Sigma + F + B\lambda \tag{4.1.8}$$

with $\Sigma = \begin{pmatrix} \sigma \\ \dot{\sigma} \end{pmatrix}$, $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 0 \\ G\Psi & \beta G\Psi \end{pmatrix}$ and $F = \begin{pmatrix} 0 \\ \Phi \end{pmatrix}$. We discretize the nonlinear terms Φ and Ψ using the explicit Euler scheme: we consider that $\Phi(t) = \Phi_k \coloneqq \Phi(t_k)$ and $\Psi(t) = \Psi_k \coloneqq \Psi(t_k)$ for $t \in [t_k, t_{k+1})$. For the last step in the discretization of (4.1.8), we use the ZOH method, which yields

$$\widetilde{\Sigma}_{k+1} = A^*\Sigma_k + F_k^* + B_k^*\lambda, \tag{4.1.9}$$

with $A^* \coloneqq e^{Ah} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$, $B_k \coloneqq G\Psi_k \begin{pmatrix} 0 & 0 \\ 1 & \beta \end{pmatrix}$ and $B_k^* \coloneqq \displaystyle\int_{t_k}^{t_{k+1}} e^{A\tau} B_k \mathrm{d}\tau = hG\Psi_k \begin{pmatrix} h/2 & \beta h/2 \\ 1 & \beta \end{pmatrix}$, $F_k \coloneqq \begin{pmatrix} 0 \\ \Phi_k \end{pmatrix}$ and $F_k^* \coloneqq \displaystyle\int_{t_k}^{t_{k+1}} e^{A\tau} F_k \mathrm{d}\tau = \begin{pmatrix} h^2\Phi_k/2 \\ h\Phi_k \end{pmatrix}$. Hence we get the system

$$\widetilde{\sigma}_{k+1} = \sigma_k + h\dot{\sigma}_k + \frac{h^2}{2}\Phi_k + \frac{h^2}{2}G\Psi_k[\lambda_1 + \beta\lambda_2] \tag{4.1.10}$$

$$\widetilde{\dot{\sigma}}_{k+1} = \dot{\sigma}_k + h\Phi_k + hG\Psi_k[\lambda_1 + \beta\lambda_2] \tag{4.1.11}$$

$$-\lambda_1 \in \mathrm{Sgn}(\widetilde{\sigma}_{k+1}) \tag{4.1.12}$$

$$-\lambda_2 \in \mathrm{Sgn}(\widetilde{\dot{\sigma}}_{k+1}), \tag{4.1.13}$$

with unknowns $\lambda_1, \lambda_2, \widetilde{\sigma}_{k+1}$ and $\widetilde{\dot{\sigma}}_{k+1}$. Note that this composite discretization procedure differs from discretizing the dynamics (4.1.8) explicitly (or with $\vartheta = 0$ as presented in Section 3.2 and 3.5). Let us check how the results from Chapter 2 can be applied on this closed-loop system. It follows from Lemma 2.3.1 that this system has always a solution. Uniqueness properties are given by Lemma 2.3.8: the control input u_k is uniquely defined as well as λ if $\widetilde{\sigma}_{k+1} \neq 0$. We solve the AVI associated to the twisting controller by enumeration: the code is given in Appendix C.2.

4.1.2 Experimental Results

This section is devoted to the analysis of the experimental results obtained on the electropneumatic setup. Recall that the control objective is to make the position of the piston track a sinusoidal trajectory. In the following, the desired trajectory is

$$y_d \coloneqq A_{\mathrm{mpl}} \sin(0.2\pi t).$$

The controller was implemented as a Simulink model and then transferred onto a DS1005 dSpace board. We were able to get results with the sampling period h in the range $[3, 100]$ms and with the gain G in the range $[10^{-2}, 10^{7}]$. The sliding surface parameter α and the two filtered differentiator time constants in (4.1.1) require proper tuning for each sampling period. They can drastically alter the performances of the controller. Since it appears that both have to be tuned together, preliminary values were obtained using simulations, with a selection based on the average error or precision, and were later refined on the plant. Section 4.1.3 is dedicated to the tuning of those parameters and to the analysis of their influences.

We now present results for two criteria: the tracking accuracy and the chattering magnitude on both the input and the output. In each case, we first compare the explicit and implicit methods, before analyzing in more depth the performances of the implicit method.

Tracking accuracy

The tracking error $e = y - y_d$ is the quantity we aim to minimize through the twisting controller. Due to the high relative degree of the system, the controller does not bring e to 0 in finite time, but rather $\sigma = \alpha e + \dot{e}$. Once the sliding phase $\sigma = 0$ occurs, the convergence of e to 0 is then exponentially fast if $\alpha > 0$. The latter parameter controls the speed of convergence: the bigger α is, the faster the error decreases.

To measure the accuracy of the tracking, we compute the average of the absolute value of the error over an interval of 60s. We call this quantity the precision and we denote it $\bar{e}$. Its analytical formula is

$$\bar{e} := \sum_{k=1}^{N} \frac{|e(t_k)|}{N} \quad \text{with} \quad t_N - t_1 = 60\text{s}. \tag{4.1.14}$$

On Figure 4.3, the precision with both the implicit and explicit controllers is displayed for different sampling periods. The implicitly discretized controller clearly yields a better performance than the explicit one, for each sampling period where the comparison is possible. Indeed it was not possible to get reliable data for large sampling periods with the explicit controller, since the plant was becoming unstable. The precision appears to increase linearly with h, or in other word it is in $O(h)$. This is underscored by the linear regression plotted on Figure 4.3. This may be surprising since we use a second-order sliding mode controller and the order should be $O(h^2)$. However, recall that $\sigma = \alpha e + \dot{e}$, with

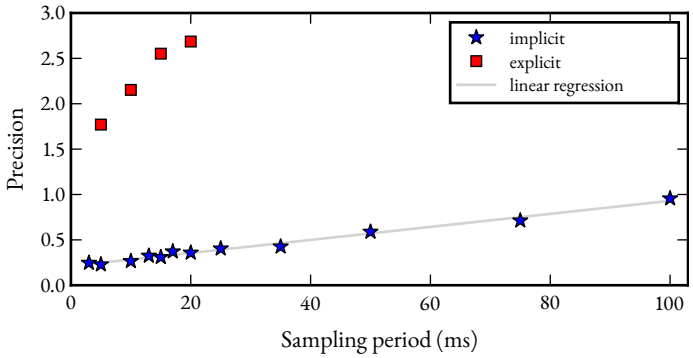
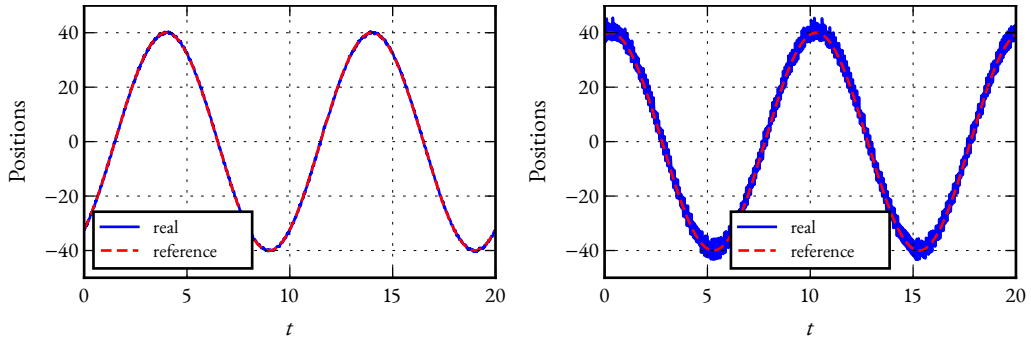


Figure 4.3: Evolution of the precision $\bar{e}$ with respect to the sampling time for both implicit and explicit discretizations. The gain used in every capture was $G = 10^5$.

the derivative being computed by a simple filtered differentiator. Looking at the C code generated using the Real-Time Workshop Toolbox, we can see that the approximated derivative $\tilde{v}$ of y is computed as the output of the following LTI system:

$$\begin{cases} a_k &= Aa_{k-1} + y_k \\ \tilde{v}_k &= Ca_k + Dy_k \end{cases},$$

with $A = -\tau^{-1}$, $D = \tau^{-1}$ and $C = -\tau^{-2}$, τ being the time constant in (4.1.1). This one-step approximation is of order h. Hence in (4.1.10), the term σ_k is known with a precision only in $O(h)$, which can be seen as a non-matching perturbation. Most of the time, in this tracking problem, the control action tries to bring $\widetilde{\sigma}_{k+1}$ to 0, see Figure 4.12 at the end of this section. The equation (4.1.10) is then the one used for the computation of the control input, propagating the error. This problem might be alleviated by the use of another differentiator, like the one proposed in [74].



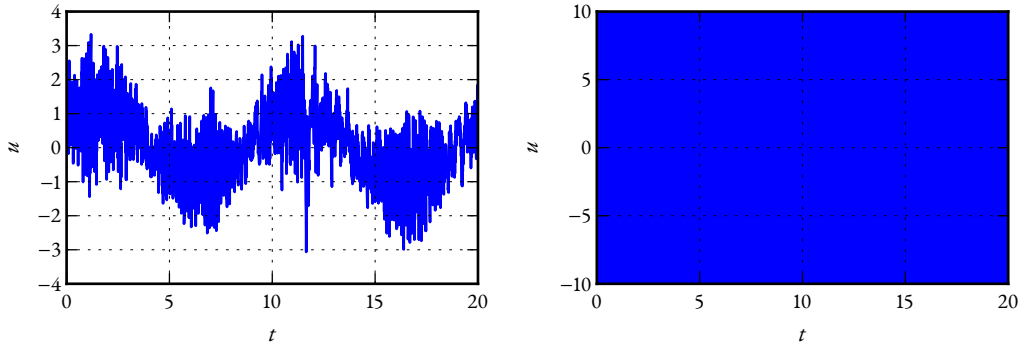
(a) With an implicit discretization.

(b) With an explicit discretization.

Figure 4.4: Real and desired position trajectories with $h = 10$ms and $G = 10^5$.

Let us show more detailed results for a specific sampling period: $h = 10$ms. On Figure 4.4a and 4.4b, the real and desired trajectories are depicted with, respectively, an

implicit and an explicit controller. On Figure 4.4a, the tracking is very accurate: at the given scale, the real position and the desired one are very close to each other. On the other hand, on Figure 4.4b, the chattering of the real trajectory is visible in the form of a boundary layer around the reference trajectory. Therefore the output chattering has been drastically reduced with the use of an implicit controller. Turning our attention to the



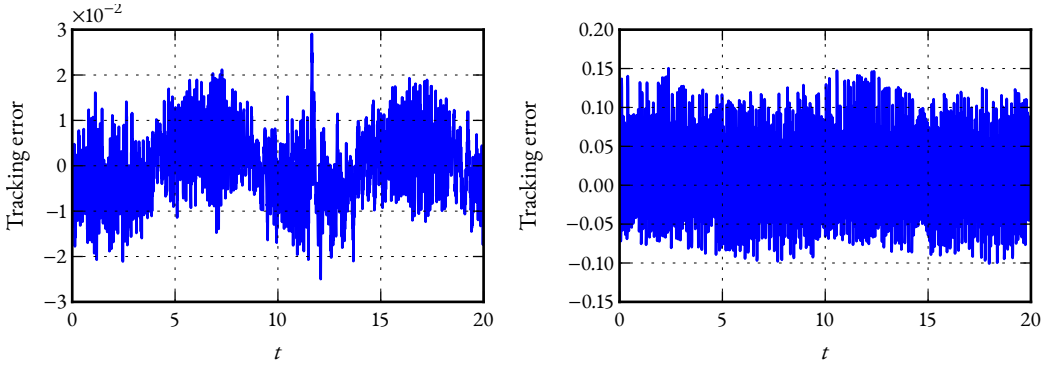
(a) With an implicit discretization. (b) With an explicit discretization.

Figure 4.5: Evolution of the control input u for both implicit and explicit discretizations with $h = 10\text{ms}$ and $G = 10^5$.

control input, Figures 4.5a and 4.5b illustrate the evolution of this quantity in the implicit and explicit cases. In the first case, the control values are in the range $[-3, 3.3]$, which is well inside the constraints $u \in [-10, 10]$. Although the control is affected by the noise from the measurements, there is an underlining periodical signal, which is also seen on simulation results, see Figure 4.13a. The root cause of the oscillations is likely to be the approximations done to get the discrete-time model in (4.1.9). It is difficult to analyze the data on Figure 4.5b since the control input is switching at a very high frequency between the 2 extremal values -10 and 10, sign of a chattering input. It is pretty clear that the main source of chattering is the explicit discretization of the controller.

Let us finish with the tracking error measured with the same two twisting controllers, as shown on Figure 4.6. Comparing the ranges, we can see that in the implicit case (Figure 4.6a), the tracking error is one order of magnitude smaller than in the explicit case (Figure 4.6b). The spike in Figure 4.6a around $t = 22\text{s}$ is due to the perturbation of the second actuator, which periodically switched its force acting on the moving mass from 1000N to -1000N and vice-versa. We further analyze the chattering in the second part of this section.

Having exposed the superiority of the implicit discretization with respect to the explicit one, let us further present the good performances that it yields. Firstly it is possible to



(a) With an implicit discretization. (b) With an explicit discretization.

Figure 4.6: Evolution of the tracking error for both implicit and explicit discretization with h = 10ms and $G = 10^5$.

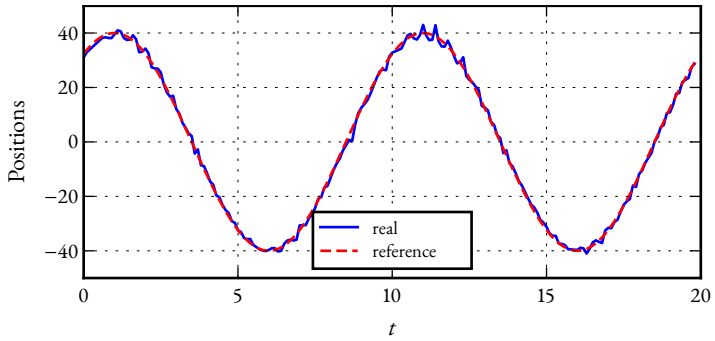


Figure 4.7: Real and desired positions with an implicit discretization, h = 100ms and $G = 10^5$.

increase the sampling period while keeping a good tracking and a system stable in practice. Figure 4.7 illustrates this fact: even with a sampling period of 100ms, the tracking takes place, although with degraded performances compared to the one in Figure 4.4a. However the precision is still better than with an explicit controller with a sampling period one order of magnitude smaller as shown in Figure 4.3 and 4.4b. Another very nice feature of

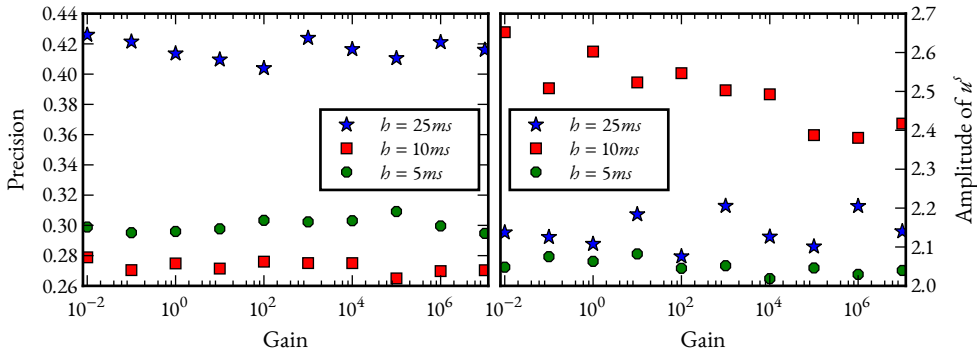


Figure 4.8: Evolution of the precision and the control input amplitude when the gain (G in (4.1.7)) varies for 3 different sampling periods.

the implicit discretization is the fact that the control input value is computed as a selection of a set-valued term, as mentioned earlier. One implication is that the gain just needs to be large enough with respect to the perturbation to ensure the robustness (remember (1.1.7)), but a further increase in the gain does not harm the performances. This is illustrated in Figure 4.8, where we display data obtained in the following way: the experiment is run with the same control 10 times, increasing the gain 10 fold each time, from 10^{-2} to 10^{7}. This was repeated for 3 different sampling periods. On Figure 4.8, both the precision $\bar{e}$ and the amplitude of the control input are plotted versus the gain G. For each sampling period, the precision varies only by less than 5%, which is solely due to the noise in the plant. The random evolution, with respect to the gain further supports this claim. Regarding the amplitude of the control input, we compute it as the mean of the top 5% values of $|u_k|$, to which were subtracted the 10 top values, as to remove any outlier. Again, we see only random variation when the gain is increased. This is close to the result we obtained for the classical SMC in Corollary 2.2.13. To get a closer look, we have in Figure 4.9 the implicit

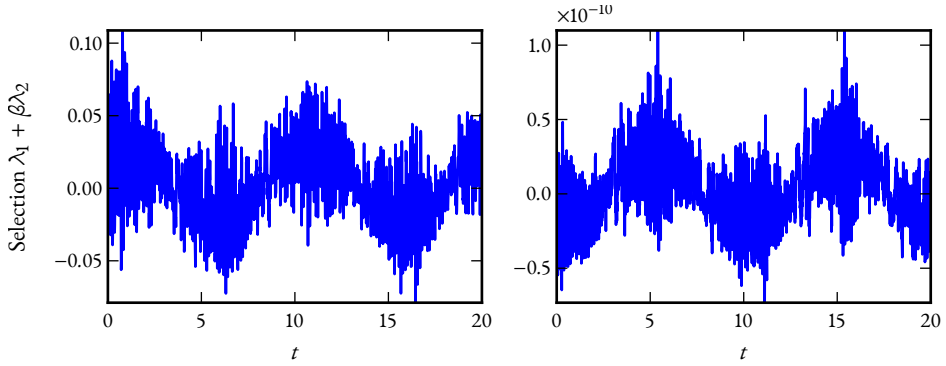


Figure 4.9: Implicit signum selections (4.1.15) for 2 values of gain: 10^{-2} and 10^{7} and with a sampling period of 10ms.

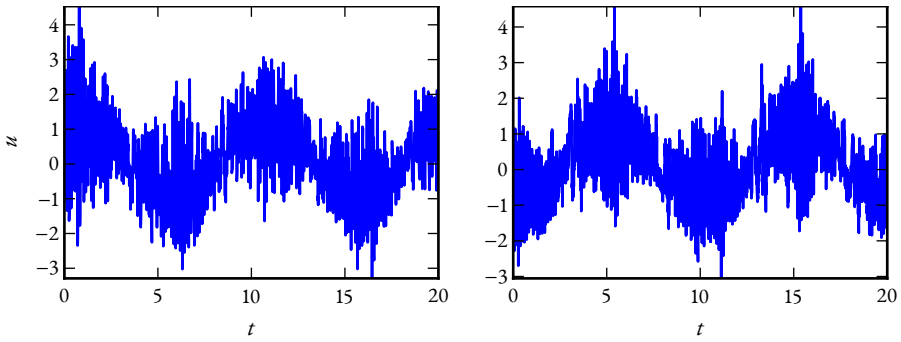


Figure 4.10: Control inputs for two values of gain: 10^{-2} and 10^{7} and with a sampling period of 10ms.

signum selections, and in Figure 4.10 the control inputs for the two extremal values of

gain: 10^{-2} and 10^{7}. We call *implicit signum selection* the quantity

$$\lambda_1 + \beta\lambda_2, \tag{4.1.15}$$

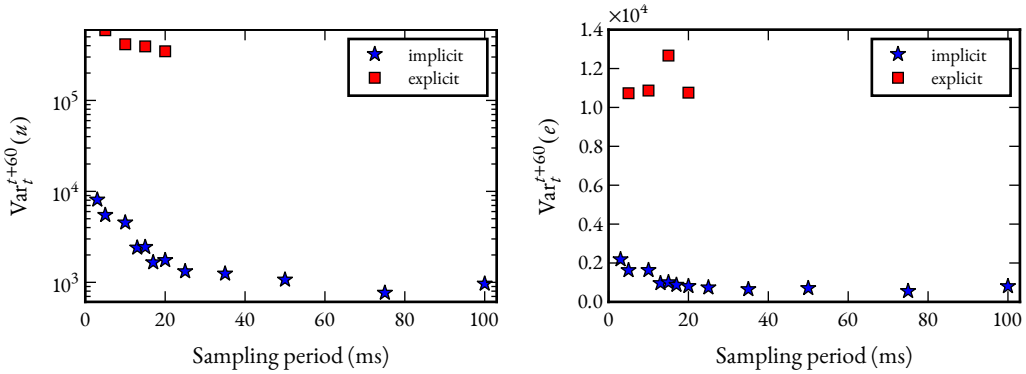
where λ_1 and λ_2 are as in (4.1.12) and (4.1.13). Multiplied by the gain G, it is equal to the control input value, see (4.1.7). The shape of the implicit signum selection is similar with both gains, however the range of the values is $[-0.08, 0.1]$ with $G_a = 10^{-2}$ whereas it is $[-0.6\,10^{-10}, 10^{-10}]$ with $G_b = 10^{7}$. The ratio between the extremal values is close to G_a/G_b. Now moving on to the control input in Figure 4.10, it does not change much: with both gains, the control input u is in the range $[-3, 4.5]$. As long as the gain G is large enough, the control input does not change much. The loose coupling between the control input and the gain is only possible with an implicit discretization, which enables us to compute the control input value as a selection. With an explicit discretization, it is well-known that the increase of the gain eventually leads to an increase in the control input and therefore an increase for both input and output chattering. The insensitivity of the discontinuous controller with respect to the increase in the gain has also been verified for the ECB-SMC controller in [112]. This is an expected property given the use of Filippov's framework. Let us switch focus on the chattering for the rest of this section.

Input and output chattering

We propose to characterize the chattering of a variable with the variation of the associated signal. Let us recall from Definition 1.3.2, the expression of the variation of a real-valued step function $f(\cdot)$ on an interval $[t, T]$:

$$\mathrm{Var}_t^T(f) := \sum_k |f(t_k) - f(t_{k-1})|,$$

with $k \in \mathbb{N}^*$ such that $t_k \in (t, T]$ and t_k are the time instants where the control input value changes. Though this quantity is not commonly used in Control Engineering, it provides a nice characterization of the chattering on either the control input or the sliding variables. We pay attention to both input and output chattering, since the first one contributes to the second and it can also induce rapid wear of actuators, especially if they are mechanical ones. Furthermore, it may also be linked to the energy consumption of the actuator. As before, we present the evolution of the control input chattering with respect to the sampling period for both implicit and explicit controllers. From Figure 4.11a, we can infer that the trend in both cases is a decrease of the variation with an increase in the sampling period. Again the implicit controller performs much better, having a control input variation two



(a) Control input variation. (b) Error variation.

Figure 4.11: Evolution of the control input variation and the error variation with respect to the sampling time for both implicit and explicit discretizations. The gain used in every capture was $G = 10^5$.

orders of magnitude smaller than the explicit one. This reduced chattering can also be assessed on site with a huge reduction of the noise made by the actuators[2].

Moving on to the output chattering, the same conclusion follows: the implicit method performs better than the explicit one, this time by an order of magnitude (see Figure 4.11b). This means that the output chattering is notably reduced. Indeed a bang-bang type control input, like the one the explicit discretization yields, tends to change the sign of the sliding variable very frequently. This leads to a large variation of the error, with respect to the variation with an implicit controller. At the same time, this behavior does not yield a better tracking, as illustrated by Figure 4.3.

Let us finish with an analysis of the values taken by λ_1 and λ_2, defined respectively in (4.1.12)

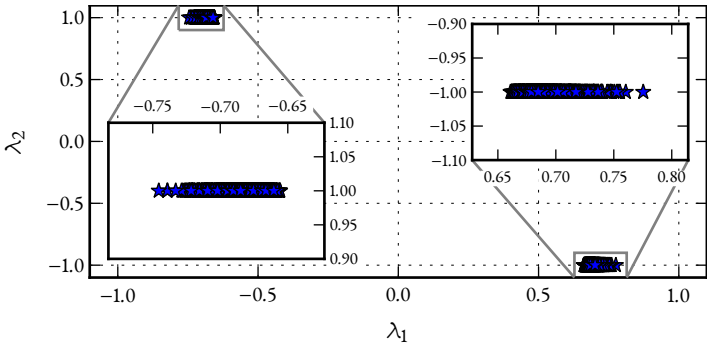


Figure 4.12: Values of the two variables λ_1 and λ_2, as defined in (4.1.12) and (4.1.13), with $G = 10^{-2}$ and $h = 10$ms.

and (4.1.13). They are the selections of the set-valued inputs, that is the real values taken

[2]The reader is invited to watch the videos at http://bipop.inrialpes.fr/people/huber/

by the controller. On Figure 4.12, we can see that λ_2 is equal to either 1 or -1 whereas λ_1 takes value in $(-.745, -.656) \cup (0.660, 0.776)$. Then from equations (4.1.10) to (4.1.13) we deduce that $\widetilde{\sigma}_{k+1} = 0$ and $\widetilde{\dot{\sigma}}_{k+1} \neq 0$. Therefore the control action tries to bring $\sigma(t_{k+1})$ to 0 at each time instant t_k. Based on this observation, one sees that (4.1.10) is in fact the equation giving its value to the implicit selection $\lambda_1 + \beta\lambda_2$. This explains the propagation of the error in the computation of $\dot{e}$ from σ_k to the control input mentioned in Section 4.1.2. It also provides an heuristic for the computation of the control: after a short period of time the tracking takes place and then the controller always brings $\widetilde{\sigma}_{k+1}$ to 0. Hence if we solve the AVI given in (4.1.11) by enumeration, we can firstly try the two cases where $\widetilde{\sigma}_{k+1} = 0$.

4.1.3 Parameters selection

We mentioned at the beginning of Section 4.1.2 that the tuning of the sliding surface parameter α and of the two filtered differentiator (τ_v and τ_a) is important and drastically affects the closed-loop behavior. In this section, we present the procedure to get initial values for those parameters and then analyze their influences on the performances. First let us recall some basic facts about the parameters we deal with. In continuous time, the sliding surface parameter α influences the error dynamics once the origin is reached. In this case, the ODE in (4.1.3) becomes $\alpha e + \dot{e} = 0$ and the exponential decrease is controlled by the value of α. Therefore we expect the performances to improve with higher values of α. Regarding the filtered differentiators, the constants on the low-pass filter should be tuned such that the dynamics of the closed-loop system are preserved as much as possible and that a major part of the measurement noise is rejected. It looks reasonable to assume that when the sampling period decreases, the high frequency part of the dynamics is richer since the control input changes more frequently. Hence, we expect the optimal value of those coefficients to decrease with the sampling period. This search for parameters is motivated by the fact that even in simulation, the closed-loop system with the implicit controller is giving good results only for a small range of sampling periods with chattering appearing suddenly with a small change in the sampling period, which is not at all consistent with the theory presented in Section 1.1.2 and 4.1.1. The fact that both the input and output chattering imputable to the controller have been greatly reduced enabled us to see the influence of other parameters. The material presented here reflects our progress with respect to this new problem: in the first subsection, we present the method used to get initial values for the parameters using a simulation-based approach. In the second subsection, we analyze the experimental data capture while varying the sliding

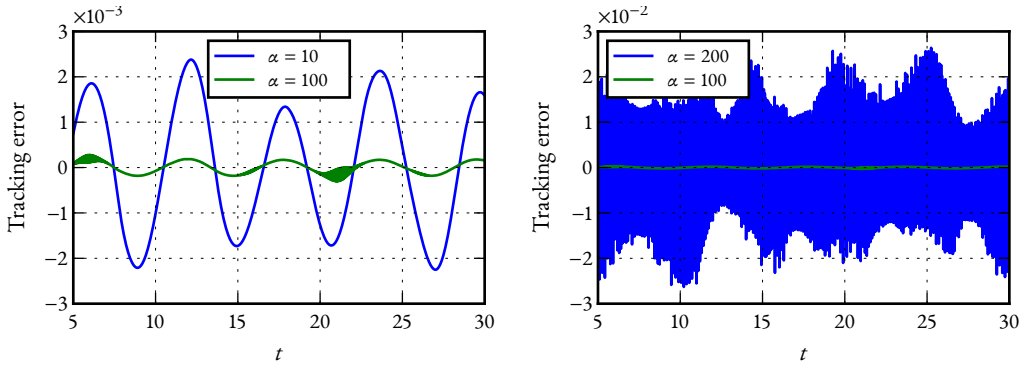
parameter α, which is part of the twisting controller. The choice of its value is one of the final step of the controller design. We hope that the material presented here and in the aforementioned reference can be used as guidelines for the selection of those parameters and help to enhance on-site fine tuning.

Simulation-based Parameters Selection

Let us recall how the parameter α was introduced: in Section 4.1.1, the sliding variable is defined as

$$\sigma = \alpha e + \dot{e} \text{ with } e = y - y_d,$$

This choice is imposed by the relative degree 3 between the tracking error e and the control input u. Let us present some simulation results illustrating the difference of behavior when α changes. On Figure 4.13a, the controller with $\alpha = 100$ is able to achieve a tracking



(a) Simulations with $\alpha = 10$ and $\alpha = 100$

(b) Simulations with $\alpha = 100$ and $\alpha = 200$

Figure 4.13: Tracking errors for two simulations with an implicit controller and $h = 15$ms. The controller parameters are the same, except the value of α.

performance that is an order of magnitude better than the controller with the sliding coefficient $\alpha = 10$. Let us underline here the fact that with an explicit discretization, it is much more difficult to see such changes: the tracking error is one order of magnitude bigger both in simulation and on the plant, see 4.3. Hence, the improvements we see on Figure 4.13a are lost in the numerical chattering. Going back to the tuning of α, this parameter value cannot be increased at will: even in simulation, some (numerical) noise is present and degrades the performances. This is illustrated on Figure 4.13b, where the value of α is doubled to 200. The tracking error is two order of magnitude higher than previously and the chattering is huge.

As mentioned in Section 4.1.1, the position y is measured but neither the speed nor the acceleration of the piston. However those quantities are required for the computation of the sliding variables σ_k and $\dot{\sigma}_k$. They are then computed using the filtered differentiator given in (4.1.1). The effects of a badly tuned filtered differentiator parameter are already visible in simulation, without purposely adding any noise. On Figure 4.14, the tracking error

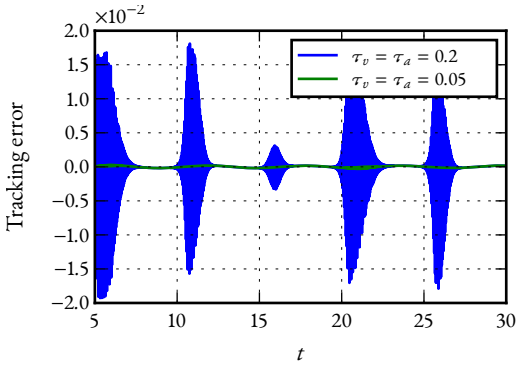


Figure 4.14: Tracking errors for two simulations with an implicit controller and $h = 15$ms. The controller parameters are the same, except the filtered differentiator parameters.

with two different sets of filtered coefficients is displayed. We reproduced the same controller as for the last simulation ($\alpha = 100$, $\tau_v = \tau_a = 0.2$) and for the second simulation, we changed the coefficients to be both equal to 0.05. It looks like a phase with an important chattering appears periodically, which considerably degrades the precision of the tracking. It quickly appeared that it was not possible to tune separately the differentiator (τ_v, τ_a) from (4.1.1) and sliding (α) coefficients. We opted for a simulation-based approach, where a range for all the coefficients is selected. We imposed the two differentiator coefficients τ_v and τ_a to be equal as to reduce the number of configurations to test. Given a sampling period, the closed-loop system runs in simulation for 30s and then we compute different metrics as the min, max and the mean absolute value of the tracking error on the position. We select only data which correspond to a simulation time in the interval $[15, 30]$s, as to eliminate any transient phase. Indeed the initial value for the physical quantities used to start the simulation may not yield a system in steady-state. This hypothesis is backed up by the fact that we sometimes see such a transient phase in simulation but we never did on the physical plant. Based on the average absolute value of the tracking error, we select which triplet of parameters should be consider as initial value for the tuning on the plant. The simulation results indicate that there are 3 orders of magnitude between the best configuration and the worst: best precision is close to $2 \cdot 10^{-4}$, whereas the worst is close to 0.1, see Figure 4.15. It also illustrate how the precision is affected by the change in the various coefficients. This numerical research of the best coefficients was performed for

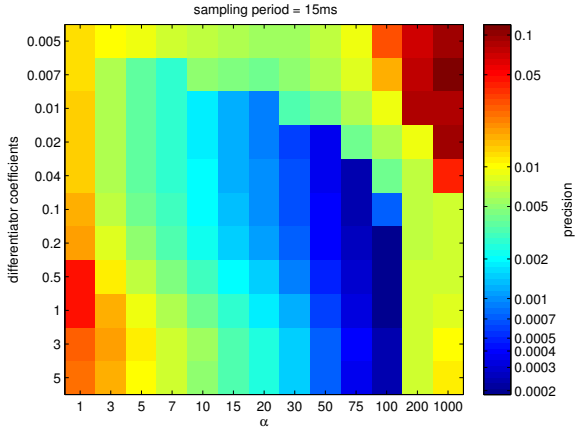


Figure 4.15: Heat map of the precision for various values of the differentiator coefficients and the sliding coefficients. The sampling period was set to 15ms.

a range of sampling periods $\{1, 3, 5, 7, 10, 15, 20, 35, 50, 75, 100\}$ms and we used those as initial values for the parameters.

Analysis of the influence of the parameter α on the experimental setup

Let us now present an analysis of some experimental data collected on the electropneumatic setup. We concentrate here only on the sliding parameter α which is part of the twisting controller. The choice of its value is one of the final step of the controller design. For the remainder of this section, all the data presented on the figures are from the experimental setup. The parameter α was tuned online by considering the noise produced by the plant, which can be linked to the actuator chattering, on this setup. Two examples of the relation

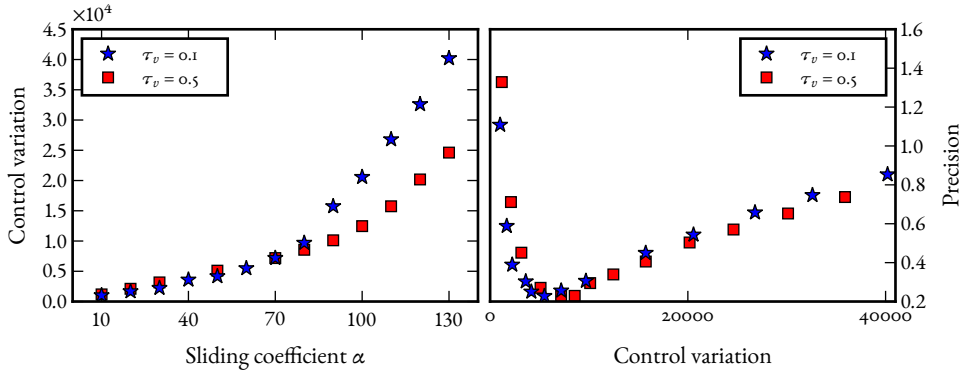
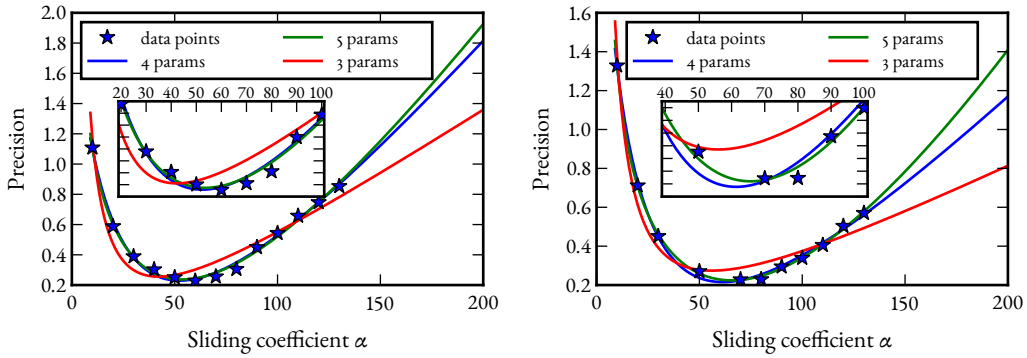


Figure 4.16: Control variation relationship with the sliding parameter α and the precision.

between the precision and the input chattering are given in Figure 4.16: after a quick improvement in the precision, the best value is obtained. Then the tracking error increases with the control input variation. It is also apparent on the left plot in Figure 4.16 that the

control variation is increasing with the sliding coefficient α. Thus a good tuning strategy is to increase α until the precision seems to deteriorate and the chattering increases. Having narrowed the interval for the optimal α, we can then try to track it.



(a) Filtered coefficient values: $\tau_v = 0.1$, $\tau_a = 0.05$. (b) Filtered coefficient values: $\tau_v = 0.5$, $\tau_a = 0.5$.

Figure 4.17: Evolution of the precision versus the sliding coefficient. The sampling period is 5ms and the filtered differentiator coefficients are the only parameters changing between the two experiments.

We continue our analysis by displaying the evolution of the precision with respect to the sliding coefficient α in Figure 4.17. We aim at a better understanding of the influence of the sliding surface on the closed-loop system performance. In the figures, we added plots of functions trying to capture the behavior of the system. To choose them, we started from two observations based on the asymptotic behavior: for small values of α, the evolution of the precision looks like $1/x$ and for the large values, the tracking error increases quasi-linearly. Therefore we tried to capture the behavior using rational functions where the degree of the numerator is the degree of the denominator plus one. We can also set the constant term of the denominator to be 0, as to impose an asymptotic behavior close to $1/x$ near 0. Nonetheless we tried with one function (f_4, see below) that does not have this property. The three functions that we tried are the following ones:

$$f_3(x) := \frac{ax^2 + bx + c}{x} \quad f_4(x) := \frac{ax^2 + bx + c}{x + d}$$
$$f_5(x) := \frac{ax^3 + bx^2 + cx + d}{x^2 + ex}.$$

The fitting of the coefficients was done in Python with the function `scipy.optimize.curve_fit` from SciPy [65], which internally calls least-square solving routines from MINPACK. If we go back to Figure 4.17, it is apparent that the curve associated with the function f_3 is not a good fit for the measures we got from the plant. With 4 or 5 parameters, the curves capture the behavior more faithfully. On Figure 4.17a the two functions give very close

results. On Figure 4.17b, the function f_5 better fits the data. To confirm this, we would need to gather more data with $\alpha > 130$.

Let us now investigate the two asymptotical behaviors as seen in Figure 4.17. Let us

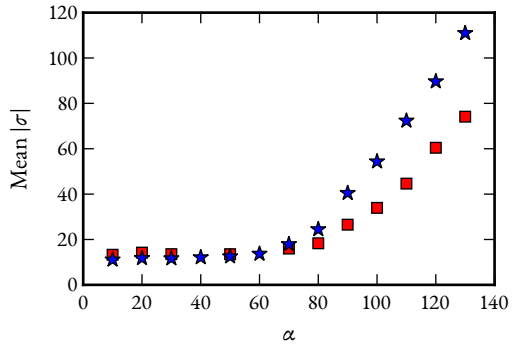


Figure 4.18: Evolution of the mean absolute value of the sliding variable σ versus the sliding coefficient.

first focus on the behavior when α is small: in this case, we can see on Figure 4.18 that the mean absolute value of σ is constant for α between 10 and 60. This is also true for the mean absolute value of the control input in Figure 4.19, and the system is in the discrete-time sliding phase since the control bounds are never hit.The system is trying to bring the sliding variable value to 0 in one sampling period. Let us formalize this observation by the following relation

$$\sigma_{k+1} = \alpha e_{k+1} + \dot{e}_{k+1} = P_k,$$

where P_k accounts for all the noise and unmodeled dynamics effects. Let us approximate $\dot{e}_{k+1}$ by $\dfrac{e_{k+1} - e_k}{h}$, which transforms the last equation into

$$e_{k+1} = \frac{e_k}{1 + \alpha h} + \frac{h}{1 + \alpha h} P_k. \tag{4.1.16}$$

This relation implies that the error e_k forms an arithmetico-geometric sequence, with the common ratio $r = P_k/\alpha$. In the following we compute the expected value of the error e_k under the hypothesis that the expected value of P_k is independent of k and is therefore denoted by $\mathbb{E}[|P|]$. The quantity we display in Figure 4.17 is $\sum_k |e_k|/N$. From (4.1.16), we get the following bounds:

$$\frac{h}{1 + \alpha h} P_k - \frac{1}{1 + \alpha h} |e_k| \leq |e_{k+1}| \leq \frac{1}{1 + \alpha h} |e_k| + \frac{h}{1 + \alpha h} P_k.$$

Summing N times this relation, ignoring the terms at the power N, and working with expectation of the error yields

$$N\frac{\mathbb{E}[|P|]}{1+\alpha+h^{-1}} - \left(|e_0| + \frac{\mathbb{E}[|P|]}{1+\alpha+h^{-1}}\right)\left(\frac{1+\alpha h}{2+\alpha h}\right) \leq \sum_k |e_k| \leq \left(|e_0| - \frac{\mathbb{E}[|P|]}{\alpha}\right)\left(\frac{1+\alpha h}{\alpha h}\right) + N\frac{\mathbb{E}[|P|]}{\alpha}.$$

Neglecting the constant terms divided by N, we finally get

$$\frac{\mathbb{E}[|P|]}{1+\alpha+h^{-1}} \leq \sum_k \frac{|e_k|}{N} \leq \frac{\mathbb{E}[|P|]}{\alpha}$$

at the limit. This small derivation corroborates the data in Figure 4.17 for small values of α, where the asymptotical behavior is like $1/\alpha$.

For large values of α, the precision looks like a linear function of the sliding parameter. This appears to be also the case for the mean absolute value of the control input, see the

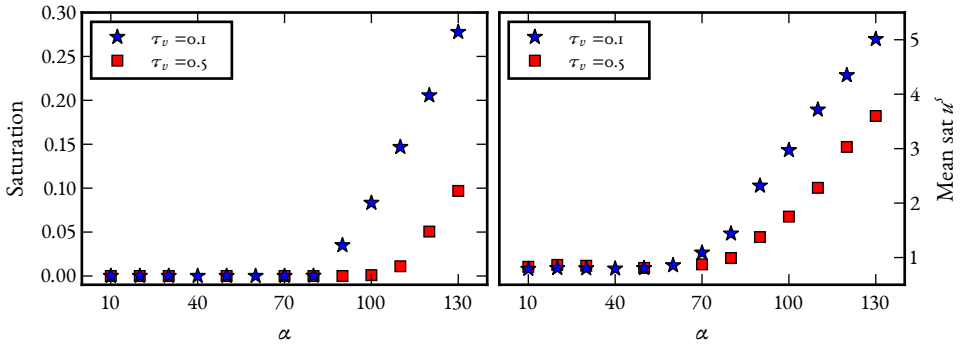


Figure 4.19: Evolution of the mean absolute value of the control input and the percentage of control input values that are saturated versus the sliding parameter α.

left plot in Figure 4.19. An interesting phenomenon is the appearance of saturation on the control input: on the right plot in Figure 4.19, for values of α greater than 90 (or 110 for the other differentiator parameters), u is hitting its bound, and it occurs more frequently when α increases. Hence we infer that the definition of the sliding variable σ can greatly affect the behavior. This is linked to the loss of homogeneity in the discrete-time controller, which is far beyond the scope of this paper. It is noteworthy that this degradation of performances was already seen in simulation, remember Figure 4.13b, where $\alpha = 100$ yields very good performance while $\alpha = 200$ yields very poor performance. Hence we suspect that this deterioration is related to the approximation on either the computation of the derivative or the discrete-time model (4.1.9). It would be of interest to see whether the use of a better

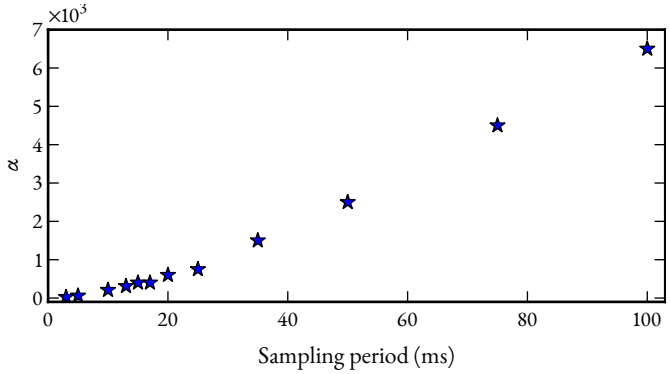


Figure 4.20: Evolution of the sliding variable parameter α versus the sampling period.

differentiator and a more accurate discrete-time model enable the use of higher values of α and if the precision is improved. Let us finish by looking at the evolution of α when the sampling period changes. The values of α presented in Figure 4.20 are those yielding the best performance in term of precision. The overall trend is an increase of the "optimal" value of α with the sampling period. With our experimental data, values are spanned from $\alpha = 25$ for $h = 3$ms to $\alpha = 6500$ for $h = 100$ms. *This important variation underlines the importance of properly tuning this sliding surface parameter when the sampling period changes.*

4.1.4 Comparison to the classical first-order sliding mode controller

Let us present some results with an implicit sliding mode controller instead of a twisting controller. For a comparison between explicitly and implicitly discretized controller for the ECB-SMC, see [113], where it is shown that the implicit controller gives much better results than the explicit one. The implicit controller in the first-order sliding mode case has the following structure:

$$-u_k \in G\,\mathrm{Sgn}(\sigma_{k+1})$$

The relative degree between the output y and the input u forces us to define the following sliding surface:

$$\sigma = \ddot{e} + 2\xi\omega\dot{e} + \omega^2 e, \tag{4.1.17}$$

with two design parameters: ξ and ω. To keep the search of the best sliding surface trackable, we fixed $\xi = 0.7$ in an analogy to the second-order ODE analysis, to theoretically ensure the fastest convergence within a 5% boundary layer of the sliding manifold. The value of ω was then tuned online to provide the best performances, see Figure 4.21. A good tuning is also instrumental in getting good results, however we do not discuss this in depth: the

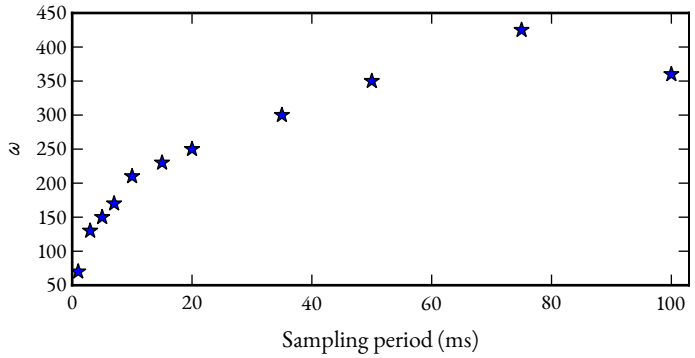


Figure 4.21: Evolution of the best value of ω versus the sampling period for an implicit sliding mode controller.

same procedure as presented for the twisting controller in Section 4.1.3 has been applied.

The control scheme is as follows: the sliding surface (4.1.17) has the dynamics

$$\dot{\sigma} = y^{(3)} - y_d^{(3)} + 2\xi\omega\ddot{e} + \omega^2\dot{e},$$

which leads to dynamics close to the twisting case (4.1.6). The nominal version of those is

$$\dot{\sigma} = \Phi' + \Psi u,$$

where $\Phi' := y^{(3)} - \Psi u + 2\xi\omega\ddot{e} + \omega^2\dot{e}$ and Ψ is the same as in (4.1.5). The discrete-time model is then derived using the same procedure as for the twisting controller. The nonlinear terms are approximated by constant terms over $[t_k, t_{k+1}]$ and hence we get the system

$$\begin{aligned} \sigma_{k+1} &= \sigma_k + h\Phi'(t_k) + h\Psi(t_k)u_k \\ -u_k &\in \mathrm{Sgn}(\sigma_{k+1}), \end{aligned}$$

which turns out to be also an equivalent form of an AVI. Therefore the existence and uniqueness properties of the control input value can be check by using the tools from Section 2.2.1. For the implementation of the controller, we used the one given in C.1, which turns out to be very simple since the sliding variable is scalar.

The resulting performances, again in term of the precision (4.1.14) are displayed in Figure 4.22, alongside the results obtained with the implicit twisting controller. As with the latter, the controller is able to provide good performance. The relationship between the precision and the sampling period appears to be linear. The implicit twisting and the classical SMC controllers yield very similar results on this experimental setup. Amongst the differences, one of the most prominent was the tuning of the sliding surface. Indeed in Figure 4.22, for a sampling period of 1ms, the closed-loop system with the implicit twisting

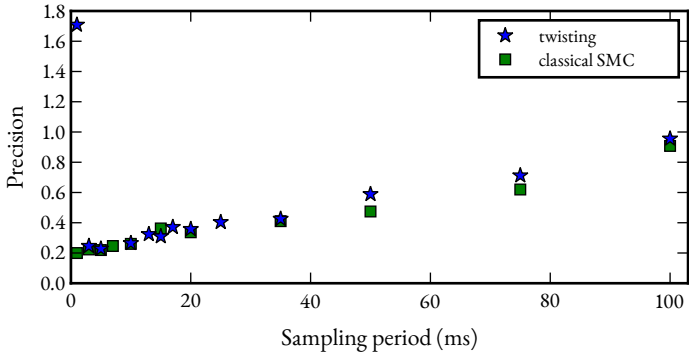


Figure 4.22: Comparison of the precision with the implicit twisting and the implicit sliding mode controller, for sampling periods in the range $[1, 100]$ms.

controller performs poorly. The precision is one order of magnitude worse than with the sampling period 3ms. This is in our opinion due to the fact that we could not find a good set of parameter values (τ_v, τ_a, α) such that the system behaves well. The behavior of the closed-loop system was similar to the case where the parameters were not properly set. This illustrates the fact that with the twisting controller, the online tuning of the parameters was getting harder as the sampling period h decreased, to the point that we failed to tune them for $h = 1$ms. For the same sampling periods, the tuning with the implicit classical sliding mode controller was much easier. However for the largest sampling periods, the situation was reversed: the implicit classical SMC controller was harder then with the twisting-based controller.

The data presented here have to be put into perspective: the performance of the closed-loop system is usually limited by the weakest component in the control loop. *Our interpretation of the results obtained from this experiment is that the "limiting" component is not the controller, but rather the ones that generate the data used to feed it, like the filtered differentiators and the linearization scheme. Enhancing those part of the controller scheme may yield better performance and might enable us to see a clear difference between the two controllers.*

4.2 Inverted Pendulum

The system, on which the experiments were conducted, is an inverted pendulum on a cart, located in the laboratory LAGIS, École Centrale de Lille, France. Only the classical, first order SMC was implemented on this setup. As with the previous experiments, a lot of effort has been put in the tuning of the filtered differentors. Again we can see the enhanced performances given by the implicitly discretized controller versus the explicit one in both

tracking accuracy and the chattering. We were able to get good results for the sampling period in a given range. In the last part we present two limitations of the plant that can explain this limited range.

4.2.1 Experimental setup

Plant dynamics, actuators and sensors

The setup consists in a linear motor to which the pendulum is fixed. The mechanism

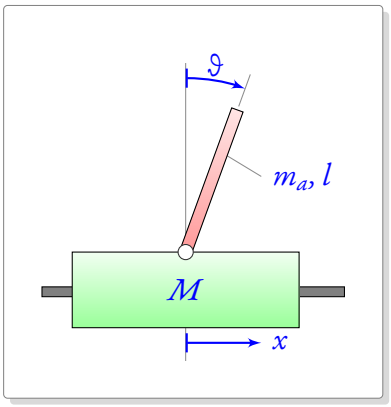


Figure 4.1: Inverted pendulum on a cart.

is sketched on Figure 4.1. We use the following linearized model around the unstable equilibrium $x_{\text{eq}} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T$:

$$\begin{cases} \dot{\mathrm{x}} = A\mathrm{x} + Bu \\ y = H\mathrm{x} \end{cases} \quad \text{with } \mathrm{x} = \begin{pmatrix} x \\ \dot{x} \\ \vartheta \\ \dot{\vartheta} \end{pmatrix}, \tag{4.2.1}$$

$$\text{and} \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{m_a}{M}g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m_a)}{Ml}g & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 \\ \frac{a}{M} \\ 0 \\ -\frac{a}{Ml} \end{pmatrix}, \; H^T = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

The masses of the cart and the pendulum are $M = 3.9249\text{kg}$ and $m_a = 0.2047\text{kg}$. We denote by $l = 0.2302\text{m}$ the length of the pendulum, $g = 9.81\text{m/s}^2$ the gravitational constant and $a = 25.3$ the motor gain. The scalar control input u is proportional to the input voltage of the linear motor. The use of the linearized model contributes to the uncertainties as well as the friction between the cart and the rod of the linear motor. It

seems reasonable to qualify those uncertainties as matched, which is backed up by the small magnitude of the sliding variable during the experiments.

Regarding the input and output of the system, the control variable u is constrained to take values between -1 and 1. The position x and the angle ϑ are available but both the speed v and angular velocities v_ϑ are computed using a filtered differentiator of the form

$$D(s) = \frac{s}{1+\tau s}$$

in the frequency domain.

Control strategy

The control objective is to maintain the pendulum at the unstable equilibrium x_{eq}. The sliding variable is scalar since there is only one control input. Since we use a simplified linear model (4.2.1) of the setup, the ZOH scheme is applied to get the following discrete-time dynamics

$$x_{k+1} = A^* x_k + B^* u_k, \quad \text{with} \quad A^* := e^{Ah} \text{ and } B^* = \int_0^h e^{A(h-\tau)} \mathrm{d}\tau B.$$

We are looking for a matrix $C \in \mathbb{R}^{1\times 4}$ to define the sliding variable σ. We use the LMI-based procedure given in [55] for relay systems to compute C. The matrix we used to get the experiments was $C = (1.38050, 1.35471, 4.13410, 0.62497)$. The computation of the control input is easy since the sliding variable is scalar: its dynamics is given by

$$\sigma_{k+1} = CA^* x_k + CB^* u_k \quad \text{and we have the relation} \quad -u_k \in \operatorname{Sgn}(\sigma_{k+1}).$$

We rewrite this as the scalar generalized equation

$$0 \in CA^* x_k + CB^* u_k + N_{[-1,1]}(u_k).$$

The computation of the control input is easy since this AVI has a scalar unknown: we can use the first technique from Section 3.1, that is

$$u_k = -\Pi_{[-1,1]} \frac{CA^* x_k}{CB^*}. \tag{4.2.2}$$

The Matlab code used to implement the implicit sliding mode controller is given in Appendix C.1.

4.2.2 Experimental results

This section is devoted to the analysis of the experimental results obtained on the inverted pendulum setup. The experiments were done with an initial position close to the unstable equilibrium in order to avoid the additional complexity of a switching logic between a local sliding mode controller and global controller. Therefore the reaching phase is short or nonexistent and the closed-loop system is mostly in the discrete-time sliding phase.

Tracking accuracy

Let us start with a comparison between two controllers which differ in the way the signum

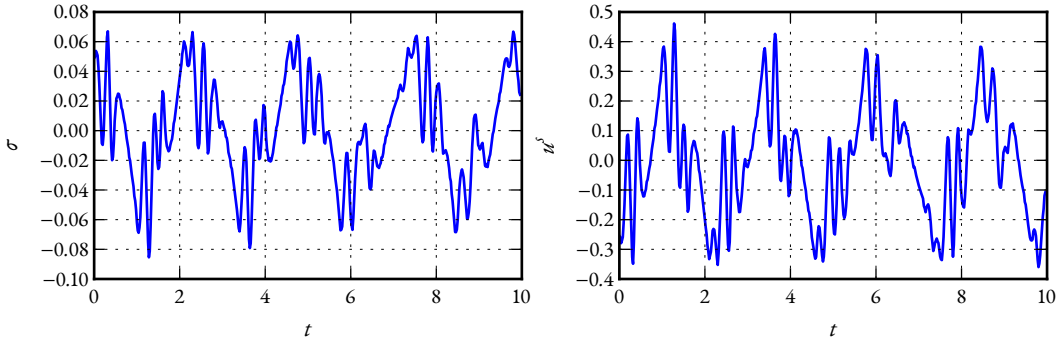


Figure 4.2: Experiments: implicit controller with $h = 20\text{ms}$, $\alpha = 1$.

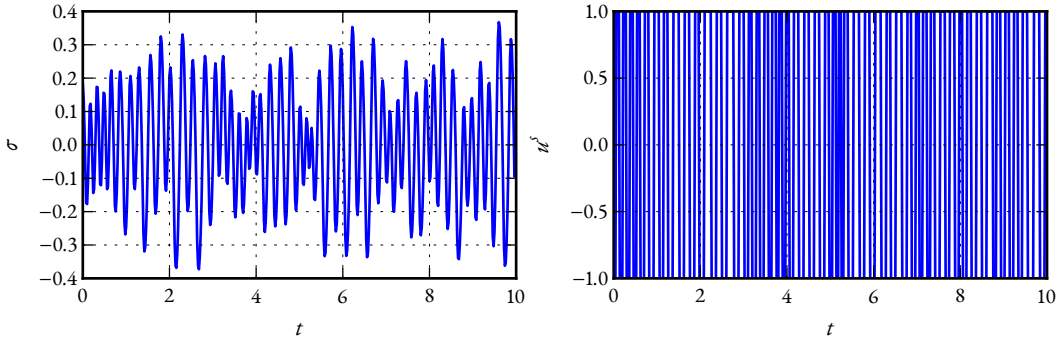


Figure 4.3: Experiments: explicit controller with $h = 20\text{ms}$, $\alpha = 1$.

function was discretized: one was implicitly discretized (Figure 4.2) and the other one explicitly (Figure 4.3). In each case, the sliding variable and the discontinuous control input u^s are depicted. With the sampling period set to 20ms, the scalar CB^* is equal to 0.1978, meaning that all the results from Section 2.2 hold. Looking at the amplitudes of the sliding variable σ in Figure 4.2 and Figure 4.3, it is clear that the implicitly discretized controller is able to maintain the value of the sliding variable an order of magnitude smaller than

the explicitly discretized one. Looking at the control input, the difference is even more striking: on Figure 4.2, the control input takes values that are proportional to the sliding variable and the control bounds are never reached, whereas in the explicit case Figure 4.3, it is a high-frequency bang-bang input.

Input and output chattering

As with the analysis of the first experimental setup in Section 4.1, we propose to characterize the chattering of a variable with the variation of the associated signal. Remember from Definition 1.3.2 that the variation of a step function $f(\cdot)$ on $[t, T]$ is defined as

$$\mathrm{Var}_t^T(f) := \sum_k |f(t_k) - f(t_{k-1})|, \tag{4.2.3}$$

with $k \in \mathbb{N}^*$ such that $t_k \in (t, T]$. We pay attention to both input and output chattering, since the first one contributes to the second and it can also induce rapid wear of actuators, especially if they are mechanical ones. Furthermore, it can also be linked to the energy consumption of certain types of actuator. The value of the variations for the data of Figure 4.2 and Figure 4.3 are displayed in Table 4.1. In both cases the implicit discretization reduces the variation (or chattering) by one order of magnitude.

Controller	$\mathrm{Var}_0^{10}(u)$	$\mathrm{Var}_0^{10}(\sigma)$
Implicit	96.24	3.10
Explicit	1332.89	44.74

Table 4.1: Control input and sliding variable variations with both the implicit and explicit controller.

In Corollary 2.2.13, we stated that with the implicit controller in the discrete-time sliding phase, the control input value does not change if the gain is increased. On Figure 4.4, this property is verified: the gain is increased threefold, but the bounds and shape of the control input are the same.

Let us also quickly present another property of the closed-loop system with an implicit controller. Figure 4.5 illustrates the good behavior of the controller when the sampling period is increased. We choose to define the precision of the controller as the mean of the absolute value of the measured values of σ. The linear regression, with slope 1.27 and

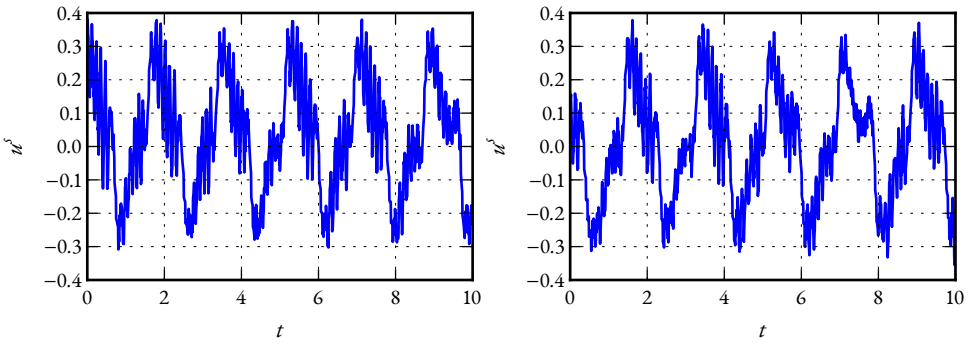


Figure 4.4: Experiments: control input values with 2 different gains: $\alpha = 1$ on the left and $\alpha = 3$ on the right; $h = 7$ms.

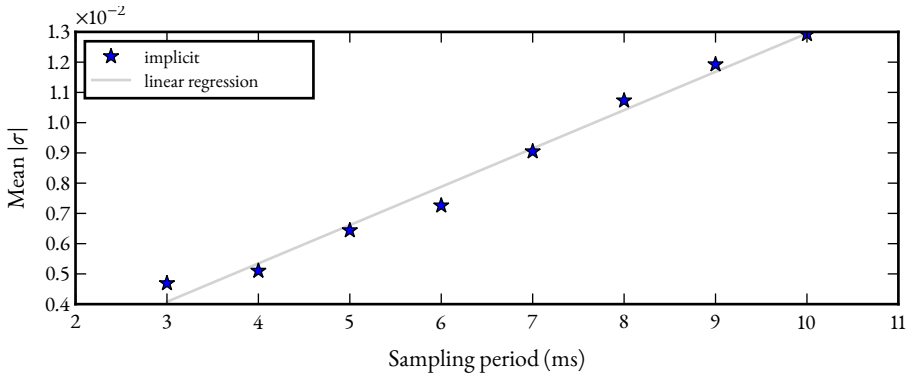


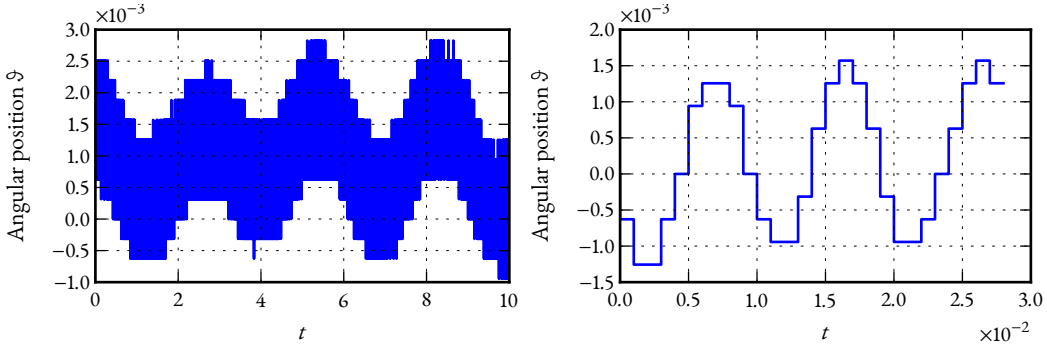
Figure 4.5: Experiments: evolution of the precision with respect to the sampling period

y-intercept $2.77 \cdot 10^{-4}$ (close to $h^2 = 4 \cdot 10^{-4}$), indicates an evolution of the precision that is linear with respect to h. Given the value of the y-intercept, it looks like the precision is $O(h)$. This is close to the result stated in Lemma 2.2.19.

This experimental data shows that the implicitly discretized controller supersedes the explicitly discretized one. Both the input and output chattering are greatly reduced. This is further investigated with the numerical simulations in the next section.

Limitation from the plant

Let us first discuss some limitations arising from the plant, which greatly limit the sampling period that can be used to get meaningful results. The first one comes from the accuracy of the sensor for the angular position ϑ. The precision of the closed-loop system increases when the sampling period decreases, which means that the angular position is closer and closer to 0. With the sampling period less than 3ms, the tracking is so accurate that the quantization arising from the accuracy of the sensor is visible, see Figure 4.6a and 4.6b. On those figures, we can see that the accuracy of the sensor is indeed $\pi/1000$. In Figure 4.6a

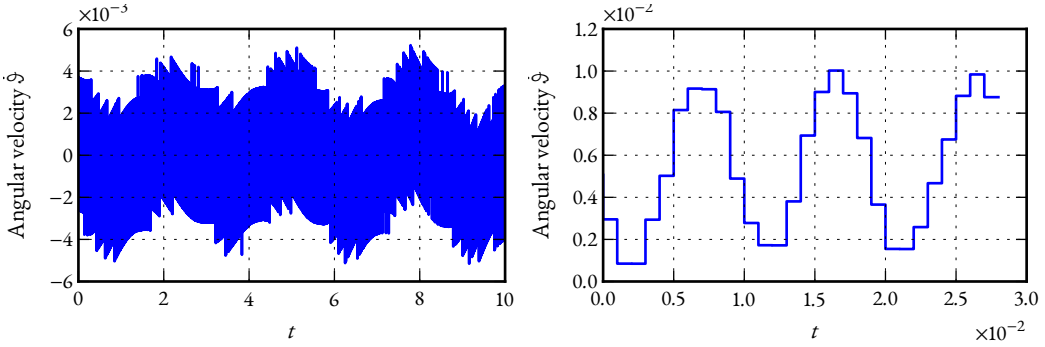


(a) The quantization is clearly visible.

(b) Detail of the left picture illustrating the behavior at the sampling period scale.

Figure 4.6: Evolution of the angular position ϑ with $h = 1$ms.

over a time interval of 10s, the sensor give only 13 different values of ϑ. This has two consequences: first the precision of the closed-loop system cannot improve much beyond even if the sampling period decreases. The second one is more problematic: the numerical derivative of this quantity has to be computed to get the value of the sliding variable σ_k. Given that we use a numerical differentiator coupled with only a first-order filter, the resulting values are not accurate and there is numerical chattering, see Figure 4.7a and 4.7b. This precludes the extended use of the implicit controller at small sampling period, since the control input has a lot of chattering, see Figure 4.8.



(a) Due to the quantization on ϑ, there is an important chattering.

(b) Detail of Figure 4.6a illustrating the behavior at the sampling period scale.

Figure 4.7: Evolution of the angular speed $\dot{\vartheta}$ with $h = 1$ms

Let us analyze further the "composition" of the chattering on the control input u. Starting from the definition of the variation in (4.2.3) and doing simple computations we have the relation

$$\mathrm{Var}_t^T(\sigma) \leq \gamma_1 \mathrm{Var}_t^T(x) + \gamma_2 \mathrm{Var}_t^T(v) + \gamma_3 \mathrm{Var}_t^T(\vartheta) + \gamma_4 \mathrm{Var}_t^T(\dot{\vartheta}),$$

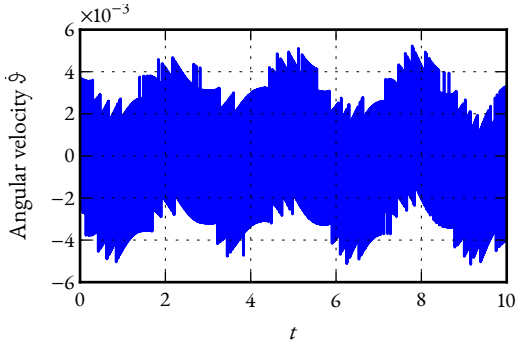


Figure 4.8: Control input with $h = 1$ms. Due to the quantization on ϑ, there is an important chattering.

with γ_i the components of the row matrix defining the sliding surface. This leads us to compute the variations of the state variables and of the control input for various sampling periods. The data is displayed in Figure 4.9 Two points are worth noting here: first the

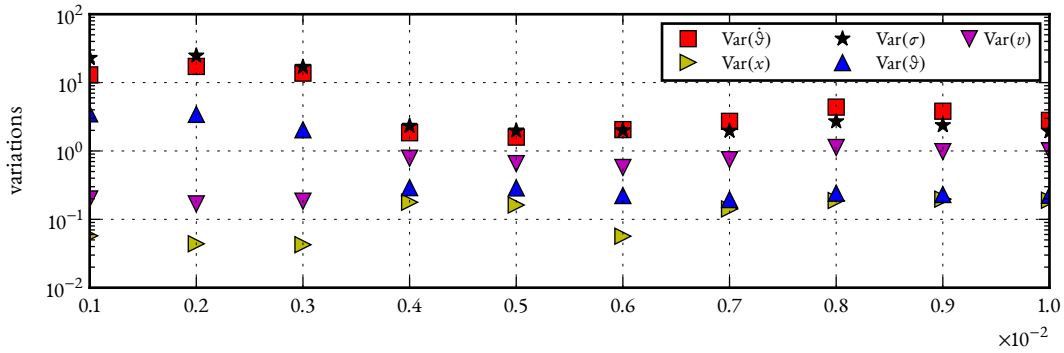


Figure 4.9: Experiments: evolution of the variations for the state and sliding variables with respect to the sampling period

control output chattering increases by one order of magnitude for sampling periods smaller than 4ms. This sudden augmentation is to be understood as a consequence of sensors accuracy issue mentioned before. This claim is backed-up by the fact that the output variation and the angular speed variation are of the same order of magnitude, whereas the other variations are all smaller. For the small sampling period the gap between the variations of u and $\dot{\vartheta}$ increases to at least one order of magnitude. Hence the behavior on small sampling periods is due to the numerical noise introduced by the sensor and its differentiation. More generally, it is also clear that the main two contributions to the sliding variable variation come from the two variables we have to numerically differentiate.

The other limitation arising from this experimental setup is as follows: there is some friction between the bar and the cart, which is not taken into account in the simple linear model used for the computation of the control input. However this nonsmooth

phenomenon is adding equilibriums when $u \neq 0$ to the closed-loop system. On Figure 4.10 this phenomenon is illustrated. With the control strategy in use, the scalar control input

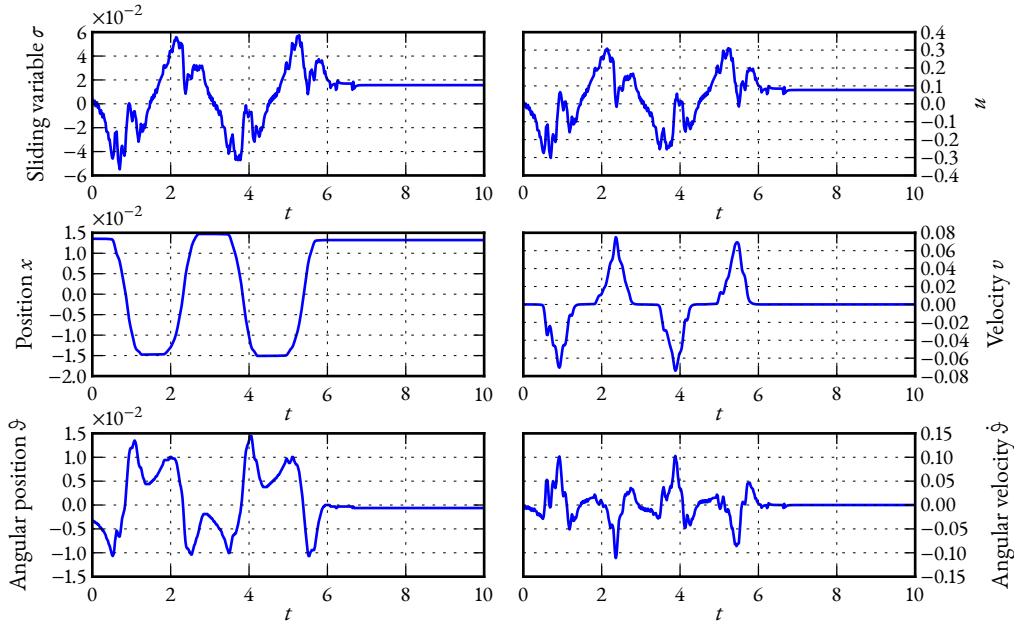


Figure 4.10: Example of an undesired stabilization by the friction

is computed as a projection, see (4.2.2). Suppose that the value of CB^* increases with h (this should hold when h is small enough). This means that for a given x_k such that $CA^*x_k/CB^* \in (-1, 1)$, an increase of h decreases the magnitude of u_k. Also the set of states such that $CA^*x_k/CB^* \in (-1, 1)$ increases with h. Hence, if at given position of the bar, the control input action for a given sampling period dominates the frictional effects, increasing the sampling period may reduce enough its magnitude such that this position may become an equilibrium of the controlled system.

Conclusion

In this chapter we presented the results of two experimental studies of discrete-time sliding mode controllers. The main objective was the comparison of the implicit and explicit versions of the sliding mode controllers. We collected data on two experimental setups: an electropneumatic system and an inverted pendulum. In the first case, extensive experiments were conducted in the context of a position tracking problem, with both the twisting and classical SMC. The analysis of the data reveals that on this electropneumatic setup, the implicit twisting controller outperforms the explicit one on three criteria: the tracking

error and both the input and output chattering. Despite the complexity of the control loop arising from the high relative degree, meaningful illustrations of theoretical results are provided, like the insensitivity with respect to an increase in the gain, once the latter is large enough. The implicit discretization allows to drastically reduce both the output and the input chattering, without modifying the controller structure compared to its continuous-time version. It also allows us to study the influence of design parameters in order to select good values. With the explicit discretization, their influence is hidden by the numerical chattering. This highlights that the implicitly discretized controller is not the "weakest" component of the control loop: other components, like the filtered differentiators, are the ones limiting the performances. This is in our view, the main reason why no significant difference between the classical SMC and twisting algorithm can be seen. On the second experimental setup, only the first-order SMC was tested. Again it performs much better than the explicit one in the three criteria. And the sampling period range for which meaningful data can be collected is also limited by two external factors: the filtered differentiator and the friction between the rod and the cart.

Conclusion

In this thesis, the discrete-time sliding mode controllers are studied. The cornerstone idea is to use the implicit discretization for the Sgn multifunction, instead of the explicit one. The main feature is that the resulting controller is still set-valued, which permits the removal of the numerical chattering phenomenon. Indeed we are able to investigate the Lyapunov stability and robustness of the closed-loop system. In many ways, the analysis is eased by the use of well-developed theories like Convex Analysis and Variational Inequalities. Those tools have already found their way in applied fields like Contact Mechanics. However we added a control theoretic flavor by relating those to classical concepts like Lyapunov functions. Also, we give an hint on how to design a controller using AVI with the modified twisting controller. Let us now summarize the main aspects of this threefold work.

ANALYSIS Expanding the ideas from [4] and [5], we further characterize the discrete-time controllers obtained through an implicit discretization of the Sgn multifunction. The main highlights in the ECB-SMC case are the Lyapunov stability analysis, the convergence of the control input to the continuous-time one, and the robustness analysis. The second important topic was the discrete-time twisting algorithm. The modified version we introduced is shown to be globally finite-time Lyapunov stable, with the help of a Lyapunov function close to the continuous-time one. We also quickly derive a discrete-time sliding mode observer, which takes the form of a reduced observer. We also investigate how one could further attenuate the effect of matched disturbances, as an indication on how to reduce the gap with the perfect attenuation in the continuous-time case.

SIMULATION The core part of the work was to implement a control toolbox in the SICONOS platform. This enables us to conduct numerical experiments complementing the theoretical analysis. The numerical results provide meaningful illustrations of theoretical results but also enable us to go beyond and provide some insights on topics that are difficult to tackle theoretically. We also underline the need for using better integration strategies

with nonlinear dynamics, such as the ϑ–γ method.

EXPERIMENTS Both the classical SMC and the twisting algorithm have been implemented on experimental setups. The main objective of those tests was to confirm that the implicit discretization of the controller outperforms the explicit one. This was verified on three criteria: the tracking error or precision, the input chattering and the output chattering. Some of the well-known properties of SMC, that were shown to hold for the implicitly discretized controller in Chapter 2, are confirmed by the analysis of the experimental data. The tests were conducted on an electropneumatic system at IRCCyN (Nantes) and on an inverted pendulum on a cart at CRIStAL (Lille).

Perspectives

The following theoretical aspects may be interesting to investigate:

- The double integrator coupled with modified twisting controller introduced in Section 2.3 was shown to be globally finite-time Lyapunov stable. This property should hold with more generic dynamics as it is the case with the continuous-time version.
- The definition of the control input value u, constrained to belong to a compact convex control set $U \subseteq \mathbb{R}^p$, as $-u \in \partial \mathfrak{h}_{-U}(\sigma)$ provides a nice starting point to generalize some of the results from Chapter 2, when the control set U is not box-shaped. In particular, the relationship between the Lyapunov function and the support function of U is worth investigating.
- The differentiators based on sliding modes like the ones introduced by Levant [74] look like a promising replacement for filtered differentiators used in the control loops for both experimental setups. Their implicit discretizations should also enhance their discrete-time implementations.

Regarding the experimental side, the following items are deemed worth looking at:

- The use of Levant's differentiator, once a study of its discretization and the resulting properties
- To implement the ϑ–γ scheme from Section 3.2. The benefits from using this method, instead of the too simple explicit Euler method, to integrate nonlinear

dynamics were already illustrated through simulations in Section 3.5. Hence, it looks interesting to implement this technique in a control loop and to test, for instance on the electropneumatic plant of Section 4.1, and see if this improves the tracking performance. The monitoring of the computational load of this method is particularly important and might limit its applicability.

Appendix A

Convex Analysis and Variational Analysis

A.1 Basics of Convex Analysis

In this Appendix, we collect basic results from Convex Analysis used in Chapter 2 for the theoretical part of this thesis. Our main references for this topic are [58, 94, 95]. We choose to expose specialized (or simplified) versions of theorems and properties since we do not need the results in their full generality. One of the peculiarities of Convex Analysis is to work with the extended real line, that is we consider functions that have an image in $\mathbb{R} \cup +\{\infty\} =: \overline{\mathbb{R}}$. Another one is the use of the epigraph, the set of points lying above the graph of a function. Let us formally define those two terms.

Definition A.1.1. The *graph* of a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is the set of points

$$\operatorname{gph} f := \left\{(x, f(x)) \mid x \in \operatorname{dom} f\right\}.$$

The *epigraph* of $f(\cdot)$ is set defined as

$$\operatorname{epi} f := \left\{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq f(x)\right\}.$$

Let us define the following assumptions (denoted by $\mathfrak{H}$) on a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$:

– $f(\cdot)$ is a convex function: for all $x, y \in \operatorname{dom} f$ and $\lambda \in [0, 1]$:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y);$$

– $f(\cdot)$ is finite everywhere ($\operatorname{dom} f = \mathbb{R}^n$), implying that f is *proper*;

- there is an affine function minorizing $f(\cdot)$ on $\mathbb{R}^n$: there exists $(s, r) \in \mathbb{R}^n \times \mathbb{R}$ such that for all $x \in \mathbb{R}^n$
$$f(x) \geq \langle s, x \rangle - r;$$
- $f(\cdot)$ is *lower semi-continuous (lsc)*, that is for all $x \in \mathbb{R}^n$, $f(x) = \liminf_{y \to x} f(y)$;
- $f(\cdot)$ is *closed*, that is its epigraph $\{(x, y) \in \mathbb{R}^n \times \mathbb{R} \mid y \geq f(x)\}$ is a closed set. If $f(\cdot)$ is convex and lsc, then it is closed.

Those assumptions hold with most of the functions used we now present and enable us to simplify the statements of the results in this appendix. Let us start with some basic definitions.

Definition A.1.2 ([58, p. 211]). The *conjugate* of a function f is the function f^* defined by
$$\mathbb{R}^n \ni s \mapsto f^*(s) := \sup_{x \in \mathbb{R}^n} \left\{ \langle s, x \rangle - f(x) \right\}.$$

The mapping $f \mapsto f^*$ is usually called the *conjugacy* operation, or the *Legendre-Fenchel* transformation.

Definition A.1.3. Let K be s subset of $\mathbb{R}^n$. The *indicator function* $\delta_K : \mathbb{R}^n \to \overline{\mathbb{R}}$ of K is defined as
$$\delta_K(x) = \begin{cases} 0 & \text{if } x \in K \\ +\infty & \text{if } x \notin K. \end{cases}$$

Definition A.1.4. The *support function* $\mathfrak{h}_K : \mathbb{R}^n \to \overline{\mathbb{R}}$ of a convex set K in $\mathbb{R}^n$ is defined by
$$\mathfrak{h}_K(x) := \sup_{y \in K} \langle x, y \rangle.$$

Fact A.1.5. The indicator function $\delta_K(\cdot)$ and the support function $\mathfrak{h}_K(\cdot)$ are conjugate of each other.

This can be easily checked from Definition A.1.2. Some useful properties of the support function are now provided.

Proposition A.1.6 ([58, p. 134]). *The support function of K is finite everywhere if and only if K is bounded.*

Proposition A.1.7 ([58, p. 137]). *For $K \subseteq \mathbb{R}^n$ nonempty, the support functions of K, $\mathfrak{h}_K(\cdot)$, and of its convex hull, $\mathfrak{h}_{\overline{\mathrm{co}}\, K}(\cdot)$, are identical.*

Example A.1.8 ($\|\cdot\|_1$ as the support function $\mathfrak{h}_{\mathcal{B}_\infty}$). First let us denote by V the set of vertices of $\mathcal{B}_\infty$. Starting with the previous property, we get that for any vector x,

$$\mathfrak{h}_{\mathcal{B}_\infty}(x) = \mathfrak{h}_V(x) = \sup_{v \in V} \langle v, x \rangle = \sum_i \max\{x_i, -x_i\} = \sum_i |x_i| = \|x\|_1.$$

Definition A.1.9. The *normal cone* to a closed convex set K is defined by

$$N_K(x) = \{d \in \mathbb{R}^n \mid \langle d, y - x \rangle \leq 0, \forall y \in K\}.$$

Fact A.1.10 ([58, p. 67]). Let K be a closed convex polyhedron defined as:

$$K = \{x \in \mathbb{R}^n \mid Hx \leq b\}, \quad \text{with } H \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m.$$

The normal cone at a point $x \in K$ is generated by the outward normals of the actives constraints:

$$N_K(x) = \left\{H_{\alpha\bullet}^T r, r \geq 0\right\},$$

with $\alpha \in \{1, \ldots, m\}$ the set of active constraints, that is for all $i \in \alpha$, we have $H_{i\bullet} x = b_i$.

Definition A.1.11 ([94, p. 214]). A vector $g \in \mathbb{R}^n$ is said to be a *subgradient* of a convex function $f(\cdot)$ at a point x if

$$f(z) \geq f(x) + \langle g, z - x \rangle, \quad \forall z \in \mathbb{R}^n. \tag{A.1.1}$$

Definition A.1.12 ([94, p. 215]). The set of all subgradients of $f(\cdot)$ at x is called the *subdifferential* of $f(\cdot)$ at x and is denoted by $\partial f(x)$. The multivalued mapping $\partial f : x \mapsto \partial f(x)$ is called the *subdifferential*.

Fact A.1.13 ([94, p. 215]). The normal cone operator $N_K(\cdot)$ is the subdifferential of the indicator function $\delta_K(\cdot)$.

Fact A.1.14. The subdifferential of $\mathfrak{h}_K(\cdot)$ at x is equal to the set of points realizing the $\sup_{y \in K} \langle y, x \rangle$:

$$g \in \partial \mathfrak{h}_K(x) \quad \Longleftrightarrow \quad g \in \arg\sup_{y \in K} \langle y, x \rangle.$$

This follows from the definition of a subgradient: take $z = 0$ in (A.1.1), which gives us

$$0 \geq \sup_{y \in K} \langle y, x \rangle - \langle g, x \rangle \quad \Longleftrightarrow \quad \langle g, x \rangle \geq \sup_{y \in K} \langle y, x \rangle.$$

The fact that

$$\forall z, \quad \mathfrak{h}_K(z) \geq \langle g, z \rangle$$

enables us to conclude.

Theorem A.1.15 (Chain Rule [94, p. 225]). *Let $f := g \circ \mathcal{A}$ with $g \colon \mathbb{R}^n \to \mathbb{R}$ a function satisfying assumptions $\mathfrak{R}$ and $\mathcal{A}$ a linear transformation from $\mathbb{R}^m$ to $\mathbb{R}^n$. Then for all $x \in \mathbb{R}^m$,*

$$\partial f(x) = A^T \partial g(Ax).$$

Theorem A.1.16 ([94, p. 219]). *Let $f(\cdot)$ be a function satisfying assumptions $\mathfrak{R}$. The mappings ∂f^* and ∂f are inverses in the sense of multivalued mappings:*

$$y \in \partial f(x) \quad \Longleftrightarrow \quad x \in \partial f^*(y).$$

One important application of this theorem is the case of the indicator and support functions.

Fact A.1.17. The subdifferentials of the indicator and the support function are inverses:

$$y \in \partial \mathfrak{h}_K(x) \quad \Longleftrightarrow \quad x \in N_K(y).$$

Let us specialize this to the case $K = \mathcal{B}_\infty$, the unit ball for $\|\cdot\|_\infty$. In Example A.1.8 we highlight that $\mathfrak{h}_{\mathcal{B}_\infty} = \|\cdot\|_1$. Hence, we have

$$\partial \mathfrak{h}_{\mathcal{B}_\infty}(x) = \partial \|x\|_1 = \partial \sum_i |x_i|.$$

Let us analyze the scalar case: if $x \neq 0$, then $|x|$ is differentiable with derivative $\operatorname{Sgn}(x)$, the multivalued signum function. Using (A.1.1), we can see that the only subgradient is $\operatorname{Sgn}(x)$. When $x = 0$, we get from (A.1.1) that for any $z > 0$,

$$z \geq \langle g, z \rangle \geq -z.$$

It is then easy to see that the subdifferential of $|\cdot|$ at 0 is $[-1, 1] = \operatorname{Sgn}(0)$. Therefore

$$\partial \mathfrak{h}_{\mathcal{B}_\infty}(x) = \operatorname{Sgn}(x),$$

and we conclude with the equivalence

$$y \in \operatorname{Sgn}(x) \quad \Longleftrightarrow \quad x \in N_{\mathcal{B}_\infty}(y). \tag{A.1.2}$$

Fact A.1.18. Let $S \subseteq \mathbb{R}^n$ and $x, y \in \mathbb{R}^n$. The following equivalence holds:

$$-y \in \partial \mathfrak{h}_{-S}(x) \iff -x \in N_S(y),$$

and the case where $S = \mathcal{B}_\infty$ gives:

$$-y \in \operatorname{Sgn}(x) \iff -x \in N_{\mathcal{B}_\infty}(y).$$

To see this, we start from (A.1.2) in Fact A.1.17:

$$\begin{aligned} -y \in \partial \mathfrak{h}_{-S}(x) &\iff x \in N_{-S}(-y) \\ &\iff \langle v + y, x \rangle \leq 0 && \text{for all } v \in -S \\ &\iff \langle -v - y, -x \rangle \leq 0 \\ &\iff \langle w - y, -x \rangle \leq 0 && \text{with } w = -v \\ &\iff -x \in N_S(y) && \text{since } w \in S. \end{aligned}$$

Proposition A.1.19 ([95, p. 213]). *With a closed convex set K, the normal cone and Euclidean projector (or projection mapping) Π_K onto K are related by*

$$N_K = \Pi_K^{-1} - I \quad \textit{and} \quad \Pi_K = (I + N_K)^{-1}.$$

A.2 Monotone Mappings

The main resource for the material presented here is [95, Chapter 12] for operators in finite-dimensional spaces. Similar results hold in the infinite-dimensional case, see [90] for instance. The main idea behind the concept of monotone mapping is to generalize the monotonicity property of a function, like the positive-semidefiniteness for linear mappings. The next statements are given for set-valued functions, so let us define the graph of a multiapplication.

Definition A.2.1. The *graph* of a set-valued function $\mathcal{A} \colon X \rightrightarrows Y$, with $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$, is the set of points

$$\operatorname{gph} \mathcal{A} := \{(x, v) \mid v \in \mathcal{A}(x)\}.$$

Definition A.2.2 ([95, p. 533]). A mapping $\mathcal{A} \colon \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is said to be *monotone* if it has the property that

$$\langle \bar{v} - \hat{v}, \bar{x} - \hat{x} \rangle \geq 0 \quad \text{whenever } \bar{v} \in \mathcal{A}(\bar{x}) \text{ and } \hat{v} \in \mathcal{A}(\hat{x}).$$

It is said to be *maximal* monotone if no enlargement of its graph is possible in $\mathbb{R}^n \times \mathbb{R}^n$ without destroying monotonicity: for all $(\bar{x}, \bar{v}) \in (\mathbb{R}^n \times \mathbb{R}^n) \setminus \operatorname{gph} \mathcal{A}$, there exists $(\hat{x}, \hat{v}) \in \operatorname{gph} \mathcal{A}$ such that $\langle \bar{v} - \hat{v}, \bar{x} - \hat{x} \rangle \leq 0$.

We shall now investigate the relation between convex functions and monotone mappings.

Theorem A.2.3 ([95, p. 542]). *Any function $f(\cdot)$ satisfying assumptions $\mathfrak{R}$ is convex if and only if its subdifferential $\partial f(\cdot)$ is monotone.*

Fact A.2.4 ([95, p. 543, 536]). For a closed convex set $K \neq \emptyset$, the normal cone operator $N_K(\cdot)$ and the subdifferential of the support function $\partial \mathfrak{h}_K(\cdot)$ are maximal monotone mappings.

Theorem A.2.5 ([95, Theorem 12.43, p. 556]). *If $S \colon \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ is maximal monotone with effective domain $\mathbb{R}^m$, then for any matrix $A \in \mathbb{R}^{m \times n}$, the mapping $T(x) := A^T S(Ax)$ is maximal monotone.*

This property as already been used in the control litterature, see [18] for example.

Appendix B

Analysis of SMC Modeled as an LCP

In Section 2.1.3, the existence of a solution to the LCP formulation of the auxiliary system (2.1.6) has been left out. We shall now investigate this matter: the LCP under study is:

$$\begin{aligned} w &= q_{\mathrm{LCP}} + \mathcal{M}_{\mathrm{LCP}} z \\ 0 \leq z \perp w &\geq 0. \end{aligned} \tag{B.1}$$

$$\mathcal{M}_{\mathrm{LCP}} \coloneqq \begin{bmatrix} 2\mathcal{M} & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \lambda_+ \\ \sigma_- \end{bmatrix}, q_{\mathrm{LCP}} \coloneqq \begin{bmatrix} q - \mathcal{M}\mathbb{1} \\ \mathbb{1} \end{bmatrix}, z = \begin{bmatrix} \lambda_+ \\ \sigma_- \end{bmatrix} \in \mathbb{R}^{2p} \text{ and } w = \begin{bmatrix} \sigma_+ \\ \lambda_- \end{bmatrix} \in \mathbb{R}^{2p}. \tag{B.2}$$

Recall that the matrix $\mathcal{M}_{\mathrm{LCP}}$ does not enjoy the P property due to the lower-right 0 block. Still it is easy to see that $z^T \mathcal{M}_{\mathrm{LCP}} z = 2\lambda_+^T \mathcal{M} \lambda_+$. Hence if $\mathcal{M}$ is positive semi-definite, then $\mathcal{M}_{\mathrm{LCP}}$ is also positive semi-definite. Let us suppose that in the sequel, $\mathcal{M}$ is positive definite. Now the following theorem gives us a good starting point in the study of the solvability of the LCP stated in (B.2).

Theorem B.1 ([27, p. 139]). *Let $\mathcal{M}$ be positive semi-definite. If the* LCP$(q, \mathcal{M})$ *is feasible, then it is solvable.*

An LCP$(q, \mathcal{M})$ is said to be *feasible* whenever there exists $z \in \mathbb{R}^p$ such that

$$z \geq 0 \quad \text{and} \quad q + \mathcal{M}z \geq 0. \tag{B.3}$$

A feasible vector z is a solution to the LCP if the complementarity condition $z^T(q + \mathcal{M}z) = 0$ holds. It turns out that this holds for the LCP (B.1) given that $\mathcal{M}$ is positive definite. Let us first state the following result.

Lemma B.2 ([27, p. 140]). *If $\mathcal{M}$ is a positive definite matrix, then there exists a vector z such that*

$$\mathcal{M}z > 0, \quad z > 0.$$

The second inequality in (B.3) in the case of the LCP (B.1) gives

$$q - \mathcal{M}\mathbb{1} + 2\mathcal{M}\lambda_+ + \sigma_- \geq 0 \tag{B.4}$$

$$\mathbb{1} - \sigma_- \geq 0. \tag{B.5}$$

A simple way to satisfy those constraints is to set

$$\sigma_- = 0 \quad \text{and} \quad \lambda_+ = \tfrac{1}{2}\mathbb{1} + \lambda',$$

with λ' to be determined. This choice ensures that (B.5) is satisfied, and (B.4) is then reduced to

$$q + 2\mathcal{M}\lambda' \geq 0. \tag{B.6}$$

Lemma B.2 guarantees the existence of λ' such that $\mathcal{M}\lambda' > 0$. Therefore by appropriate scaling, (B.6) holds. Hence, the LCP is feasible and by Theorem B.1 it is also solvable. Let us now study the uniqueness of λ and σ when $\mathcal{M}$ is positive definite. To see this, let us first present a characterization of the solution set of an LCP.

Theorem B.3 ([27, p. 141]). *Let $N \in \mathbb{R}^{n\times n}$ be positive semi-definite, and let $r \in \mathbb{R}^n$ be arbitrary. If* LCP(r, N) *has a solution, then* SOL(r, N) *is polyhedral and equals to*

$$\left\{ z \in \mathbb{R}^n_+ \mid r + Nz \geq 0, r^T(z - \bar{z}) = 0, N_s(z - \bar{z}) = 0 \right\}, \tag{B.7}$$

where $\bar{z}$ is an arbitrary solution of LCP(r, N) *and N_s is the symmetric part of N.*

Applying this result to the LCP (B.1), we get from the third condition (B.7) that

$$\begin{pmatrix} 2\mathcal{M}_s & 0 \\ 0 & 0 \end{pmatrix} (z - \bar{z}) = 0,$$

which implies that

$$z - \bar{z} = \begin{pmatrix} 0 \\ \star \end{pmatrix} = \begin{pmatrix} \lambda_+ - \bar{\lambda}_+ \\ \sigma_- - \bar{\sigma}_- \end{pmatrix}.$$

Hence $\lambda_+ = \bar{\lambda}_+$ and since $\lambda_+ + \lambda_- = \mathbb{1}$, we infer $\lambda = \bar{\lambda}$. Whence σ is also unique as $\mathcal{M}$ is positive definite.

Appendix C

MATLAB Code

C.1 MATLAB code for implicit scalar SMC

We present here the code used in the implementation of the implicit first order sliding mode controller in the case where the sliding variable is scalar. This code has been used to get the results from Section 4.1 and 4.2. The controller is created in Simulink inside a "Matlab function" block, with the code written in the Matlab language. It is then translated into *C* and compiled for the targeted microcontroller by the real-time workshop toolbox.

```
function u_k = fcn(G, h, s_k, CBk)

toProj = -s_k/(G*CBk*h);

if toProj > 1.0
  u_k = G;
elseif toProj < -1.0
  u_k = -G;
else
  u_k = G*toProj;
end

end
```

C.2 MATLAB code for the implicit twisting algorithm

The following code was used to implement the implicit twisting algorithm in its non-modified version. It was used to get the experimental results from Section 4.1 with the same implementation method as with the previous code.

```
function [lambda1,lambda2] = fcn(sigma, sigmaDot, K, beta, h, Psi, Phi, pL1, pL2)
%construct matrices
mat(2,:) = K*Psi*[h h*beta]; mat(1,:) = h/2*mat(2,:);
q = [sigma + h*sigmaDot + h*h/2*Phi; sigmaDot + h*Phi];
% first try the previous value
lambda = zeros(1, 2); lambda1 = 0; lambda2 = 0; lambda(1) = pL1; lambda(2) = pL2;
if abs(lambda(1)) ~= 1.0
    lambda(1) = 1.0;
end
if abs(lambda(2)) ~= 1.0
    lambda(2) = 1.0;
end

nbIter = 0; nposIdxOld = 0; idxZero = 0; nbIterMax = 9;
alreadyDone = zeros(9, 1); alreadyDone(lambda(1) + 2*lambda(2)+6) = 1;
posIdxOld = zeros(1, 2); oldLambdaV = lambda;

while nbIter < nbIterMax
% see if we can reach the origin
if (lambda(1) == 0) && (lambda(2) == 0)
  lambda = oldLambdaV; nposIdx = 1; posIdx = abs(lambda(1) + lambda(2))/2 + 1;
  posIdxOld = zeros(1, 2); nposIdxOld = 0;
else
if idxZero > 0
  if idxZero == 1 % try lambda(1) in [-1, 1]
    valU1 = -(q(1) + mat(1, 2)*lambda(2));
    if (abs(valU1) < mat(1, 1))
      lambda(1) = valU1/mat(1, 1); Sigma = q + mat*lambda';
      eps0 = abs(Sigma) < eps; Sigma(eps0) = 0.0;
      if sign(Sigma(2)) == -sign(lambda(2))
        lambda1 = lambda(1); lambda2 = lambda(2); return;
      end
    end
    lambda = oldLambdaV; nposIdxOld = 0; posIdx = abs(lambda(1) + lambda(2))/2 + 1;
    nposIdx = 1; posIdxOld = zeros(1, 2);
  else % try lambda(2) in [-1, 1]
    valU2 = -(q(2) + mat(2, 1)*lambda(1));
    if abs(valU2) < mat(2, 2)
      lambda(2) = valU2/mat(2, 2); Sigma = q + mat*lambda';
      eps0 = abs(Sigma) < eps; Sigma(eps0) = 0.0;
      if sign(Sigma(1)) == -sign(lambda(1))
        lambda1 = lambda(1); lambda2 = lambda(2); return;
      end
    end
    lambda = oldLambdaV; posIdxOld = zeros(1, 2);
    posIdx = abs(lambda(1) + lambda(2))/2 + 1;
    nposIdx = 1; nposIdxOld = 0;
  end
else % lambda is one of the vertex of K
    Sigma = q + mat*lambda'; eps0 = abs(Sigma) < eps; Sigma(eps0) = 0.0; nposIdx = 0;
    sLambda = sign(lambda)'; sProd = sLambda.*sign(Sigma); posIdx = [-1, -1];
    if sProd(1)>0
      posIdx(1) = 1; nposIdx = 1;
    end
    if sProd(2)>0
      if nposIdx == 0
        posIdx(1) = 2;
      else
        posIdx(2) = 2;
```

```
      end
      nposIdx = nposIdx + 1;
    end
    if nposIdx == 0 % if this is true, we are done
      lambda1 = lambda(1); lambda2 = lambda(2); return;
    end
  end
end

% prepare next iteration
idxZero = 0; oldLambdaV = lambda; hasZero = 0;
for ii=1:nposIdx
  idx = posIdx(ii);
  if (nposIdxOld == nposIdx) && (((nposIdxOld >= 1) ...
    && (idx == posIdxOld(1))) || ((nposIdxOld >= 2) && (idx == posIdxOld(2))))
    lambda(idx) = 0; idxZero = idx; hasZero = 1;
  elseif hasZero == 0
    lambda(idx) = lambda(idx) - 2*sign(lambda(idx));
    lambda(idx) = lambda(idx)/abs(lambda(idx));
  end
end
nbIter = nbIter + 1; nposIdxOld = nposIdx; posIdxOld(1:nposIdx) = posIdx(1:nposIdx);
if hasZero == 0 && (lambda(2) ~= 0)
  idxConfig = lambda(1) + 2*lambda(2) + 6;
elseif (lambda(1) == 0) && (lambda(1) == 0)
  idxConfig = 1;
else
  idxConfig = 6 + 2*(lambda(1)-1);
end
if (alreadyDone(idxConfig) == 0)
  alreadyDone(idxConfig) = 1;
else
  oldLambdaV = lambda;
  % find next available config
  if nbIter == nbIterMax
    break
  end
  jj = mod(idxConfig, nbIterMax) + 1;
  while 1
    if (alreadyDone(jj) == 0)
      alreadyDone(jj) = 1; break;
    else
      jj = mod(jj, nbIterMax) + 1;
    end
  end
  idxZero = 0;
  switch(jj)
    case 1, lambda = [0 0];
    case 2, lambda = [-1 0]; idxZero = 2;
    case 3, lambda = [-1 -1];
    case 4, lambda = [0 -1]; idxZero = 1;
    case 5, lambda = [1 -1];
    case 6, lambda = [1 0]; idxZero = 2;
    case 7, lambda = [-1 1];
    case 8, lambda = [0 1]; idxZero = 1;
    case 9, lambda = [1 1];
  end
end
end
```

```
end
```

Appendix D

Matrix Facts

We collect here a few facts about matrices used in proofs, for the reader's convenience.

Definition D.1. A matrix $A \in \mathbb{R}^{n\times n}$ is *normal* if

$$AA^T = A^T A.$$

Corollary D.2 ([57, Corollary 4.2.4, p. 399]). *Suppose that $I \subset \mathbb{R}$ is an interval and $A(\tau) \in \mathbb{C}^{n\times n}$ depends continuously on $\tau \in I$. Then there exist (single valued) continuous functions $\lambda_i : I \to \mathbb{C}$ such that*

$$\Lambda(A(\tau)) = (\lambda_1(\tau), \ldots, \lambda_n(\tau)), \quad \tau \in I,$$

with $\Lambda(A)$ the unordered n-tuple of the eigenvalues of A taking into account multiplicities.

Corollary D.3 ([57, Corollary 4.2.16, p. 405]). *Let $A \in \mathbb{C}^{n\times n}$ be normal and Δ be arbitrary. If μ is an eigenvalue of $A + \Delta$ and $\|\cdot\|_{2,2}$ denotes the spectral norm, then*

$$\min_{\lambda} |\lambda - \mu| \leq \|\Delta\|_{2,2},$$

with λ an eigenvalue of A.

Fact D.4 ([15, Fact 6.3.28, p. 410]). If $A \in \mathbb{R}^{n\times n}$ is nonsingular, then the following relation between singular values holds:

$$\nu_{\min}(A) = \frac{1}{\nu_{\max}(A^{-1})}).$$

Corollary D.5 ([59, Corollary 3.1.5, p.151]). *Let $A \in \mathbb{R}^{n\times n}$ be given with its ordered singular values, being*

$$\nu_1(A) \geq \ldots \geq \nu_n(A).$$

Let $A_s = \frac{1}{2}(A + A^T)$ *with its algebraically decreasingly ordered eigenvalues being*

$$\lambda_1(A_s) \geq \ldots \geq \lambda_n(A_s).$$

Then

$$v_i \geq \lambda_i(A_s) \quad \textit{for } i \in \{1, \ldots, n\}.$$

Bibliography

[1] K. Abidi, J.-X. Xu, and Y. Xinghuo, *On the discrete-time integral sliding-mode control*, Automatic Control, IEEE Transactions on, 52 (2007), pp. 709–715. (*Cited on p. 16.*)

[2] V. Acary, B. Bremond, O. Huber, and F. Pérignon, *An introduction to Siconos*, Rapport Technique RT-0340, INRIA, 2015. (*Cited on pp. 82 and 109.*)

[3] V. Acary and B. Brogliato, *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*, vol. 35 of Lecture Notes in Applied and Computational Mechanics, Springer Berlin Heidelberg, 2008. (*Cited on pp. 14 and 98.*)

[4] V. Acary and B. Brogliato, *Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems*, Systems & Control Letters, 59 (2010), pp. 284–293. (*Cited on pp. 2, 31, and 159.*)

[5] V. Acary, B. Brogliato, and Y. Orlov, *Chattering-free digital sliding-mode control with state observer and disturbance rejection*, Automatic Control, IEEE Transactions on, 57 (2012), pp. 1087–1101. (*Cited on pp. 2, 31, and 159.*)

[6] J. Ackermann and V. Utkin, *Sliding mode control design based on Ackermann's formula*, Automatic Control, IEEE Transactions on, 43 (1998), pp. 234–237. (*Cited on p. 7.*)

[7] M. A. Aizerman and F. R. Gantmakher, *On certain switching specifics in nonlinear automatic control systems with a piecewise-smooth characteristics of nonlinear element*, Automatika i Telemekhanika, 18 (1957), pp. 1017–1028. (*Cited on p. 1.*)

[8] M. A. Aizerman and Y. S. Pyatnitskii, *Fundamentals of the theory of discontinuous systems. I*, Avtomatika i Telemekhanika, (1974), pp. 33–47. (*Cited on p. 11.*)

[9] M. A. Aizerman and Y. S. Pyatnitskii, *Fundamentals of the theory of discontinuous systems. II*, Avtomatika i Telemekhanika, (1974), pp. 39–61. (*Cited on p. 11.*)

[10] L. Ambrosio, N. Fusco, and D. Pallara, *Functions of Bounded Variation and Free Discontinuity Problems*, Clarendon Press Oxford, 2000. (*Cited on p. 13.*)

[11] J. André and P. Seibert, *Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten I*, Archiv der Mathematik, 7 (1956), pp. 148–156. (*Cited on p. 1.*)

[12] J. André and P. Seibert, *Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten II*, Archiv der Mathematik, 7 (1956), pp. 157–164. (*Cited on p. 1.*)

[13] J.-P. Aubin and A. Cellina, *Differential Inclusions*, vol. 264 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag Berlin Heidelberg, 1984. (*Cited on p. 10.*)

[14] J. Bastien and M. Schatzman, *Numerical precision for differential inclusions with uniqueness*, Mathematical Modelling and Numerical Analysis, 36 (2002), pp. 427–460. (*Cited on p. 17.*)

[15] D. Bernstein, *Matrix Mathematics*, Princeton University Press, 2009. (*Cited on p. 175.*)

[16] V. Brasey, *HEM5 user's guide*, technical report, Université de Genève, 1994. (*Cited on p. 102.*)

[17] B. Brogliato, *Nonsmooth Mechanics: Models, Dynamics and Control*, Communications and Control Engineering, Springer London, 1999. (*Cited on p. 2.*)

[18] B. Brogliato, *Absolute stability and the lagrange-dirichlet theorem with monotone multivalued mappings*, Systems & control letters, 51 (2004), pp. 343–353. (*Cited on p. 168.*)

[19] B. Brogliato, O. Egeland, R. Lozano, and B. Maschke, *Dissipative Systems Analysis and Control: Theory and Applications*, Communications and Control Engineering, Springer London, second edition ed., 2007. (*Cited on p. 29.*)

[20] K. Çamlibel, *Complementarity Methods in the Analysis of Piecewise Linear Dynamical Systems*, PhD thesis, Tilburg University, 2001. (*Cited on pp. 16 and 97.*)

[21] K. Çamlibel, W. Heemels, and J. M. Schumacher, *Consistency of a time-stepping method for a class of piecewise-linear networks*, Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 49 (2002), pp. 349–357. (*Cited on p. 16.*)

[22] K. Çamlibel, W. Heemels, and J. M. Schumacher, *On linear passive complementarity systems*, European Journal of Control, 8 (2002), pp. 220–237. (*Cited on p. 16.*)

[23] M. Cao, *Piecewise Linear Homotopies and Affine Variational Inequalities*, PhD thesis, University of Wisconsin–Madison, 1994. (*Cited on p. 94.*)

[24] M. Cao and M. C. Ferris, *A pivotal method for affine variational inequalities*, Mathematics of Operations Research, 21 (1996), pp. 44–64. (*Cited on p. 94.*)

[25] S. D. Cohen and A. C. Hindmarsh, *CVODE, a stiff/nonstiff ODE solver in C*, Computers in Physics, 10 (1996), pp. 138–143. (*Cited on p. 104.*)

[26] R. Cottle, *A field guide to the matrix classes found in the literature of the linear complementarity problem*, Journal of Global Optimization, 46 (2010), pp. 571–580. (*Cited on p. 21.*)

[27] R. Cottle, J.-S. Pang, and R. Stone, *The Linear Complementarity Problem*, no. 60 in Classics in Applied Mathematics, Society for Industrial Mathematics, 2009. (*Cited on pp. 21, 22, 52, 98, 169, and 170.*)

[28] T. Donchev and E. Farkhi, *Stability and Euler approximation of one-sided lipschitz differential inclusions*, SIAM Journal on Control and Optimization, 36 (1998), p. 780. (*Cited on p. 16.*)

[29] A. Dontchev and F. Lempio, *Difference methods for differential inclusions: a survey*, SIAM Review, (1992), pp. 263–294. (*Cited on p. 16.*)

[30] S. Drakunov and V. Utkin, *On discrete-time sliding modes*, in Proc. of IFAC Nonlinear Control System Design Conf., 1989, pp. 273–278. (*Cited on p. 15.*)

[31] B. Draženović, *The invariance conditions in variable structure systems*, Automatica, 5 (1969), pp. 287–295. (*Cited on p. 7.*)

[32] B. C. Eaves, *A short course in solving equations with pl homotopies*, in Nonlinear Programming, R. W. Cottle and C. E. Lemke, eds., American Mathematical Society, 1976, pp. 73–143. (*Cited on p. 94.*)

[33] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications*, vol. 7 of Systems and Control Book Series, CRC Press, 1998. (*Cited on pp. 6, 69, 72, and 73.*)

[34] S. V. Emel'yanov, S. K. Korovin, and L. V. Levantovskii, *Higher-order sliding modes in binary control systems*, in Soviet Physics Doklady, vol. 31, 1986, p. 291. (*Cited on p. 1.*)

[35] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume I*, Springer Series in Operations Research, Springer, 2003. (*Cited on pp. 23 and 24.*)

[36] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume II*, Springer Series in Operations Research, Springer, 2003. (*Cited on pp. 98 and 101.*)

[37] A. F. Filippov, *Differential equations with discontinuous right-hand side*, Matematicheskii Sbornik, 51 (1960), pp. 99–128. (*Cited on p. 11.*)

[38] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, vol. 18 of Mathematics and Its Applications, Kluver Academic Publishers, 1988. (*Cited on pp. 1, 5, 10, and 12.*)

[39] I. Flügge-Lotz, *Discontinuous Automatic Control*, Princeton University Press Princeton, NJ, 1953. (*Cited on p. 1.*)

[40] B. Fornberg, *Generation of finite difference formulas on arbitrarily spaced grids*, Mathematics of Computation, 51 (1988), pp. 699–706. (*Cited on p. 80.*)

[41] L. Fridman, *An averaging approach to chattering*, Automatic Control, IEEE Transactions on, 46 (2001), pp. 1260–1265. (*Cited on p. 12.*)

[42] K. Furuta, *Sliding mode control of a discrete system*, Systems & Control Letters, 14 (1990), pp. 145–152. (*Cited on pp. 15, 34, and 38.*)

[43] K. Furuta and Y. Pan, *Variable structure control with sliding sector*, Automatica, 36 (2000), pp. 211–228. (*Cited on p. 16.*)

[44] K. Furuta and Y. Pan, *Discrete-time variable structure control*, in Variable Structure Systems: Towards the 21st Century, Y. X. and J.-X. Xu, eds., vol. 472 of Lecture Notes in Control and Information Sciences, Springer Berlin Heidelberg, 2002, pp. 57–81. (*Cited on p. 43.*)

[45] Z. Galias and X. Yu, *Complex discretization behaviors of a simple sliding-mode control system*, Circuits and Systems II: Express Briefs, IEEE Transactions on, 53 (2006), pp. 652–656. (*Cited on pp. 12, 16, and 110.*)

[46] Z. Galias and X. Yu, *Euler's discretization of single input sliding-mode control systems*, Automatic Control, IEEE Transactions on, 52 (2007), pp. 1726–1730. (*Cited on pp. 12 and 16.*)

[47] Z. Galias and X. Yu, *Analysis of zero-order holder discretization of two-dimensional sliding-mode control systems*, Circuits and Systems II: Express Briefs, IEEE Transactions on, 55 (2008), pp. 1269–1273. (*Cited on pp. 12, 16, 40, 110, and 111.*)

[48] W. Gao, Y. Wang, and A. Homaifa, *Discrete-time variable structure control systems*, Industrial Electronics, IEEE Transactions on, 42 (1995), pp. 117–122. (*Cited on p. 15.*)

[49] G. Golo and Č. Milosavljević, *Robust discrete-time chattering free sliding mode control*, Systems & Control Letters, 41 (2000), pp. 19–28. (*Cited on pp. 16 and 32.*)

[50] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, vol. 8 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, second revised ed., 1993. (*Cited on p. 14.*)

[51] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, vol. 14 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, second revised ed., 1996. (*Cited on p. 14.*)

[52] I. Haskara, Ü. Özgüner, and V. Utkin, *On sliding mode observers via equivalent control approach*, International Journal of Control, 71 (1998), pp. 1051–1067. (*Cited on p. 74.*)

[53] W. Heemels, J. M. Schumacher, and S. Weiland, *Linear complementarity systems*, SIAM Journal on Applied Mathematics, 60 (2000), pp. 1234–1269. (*Cited on p. 16.*)

[54] L. Hetel and E. Fridman, *Robust sampled–data control of switched affine systems*, Automatic Control, IEEE Transactions on, 58 (2013), pp. 2922–2928. (*Cited on p. 4.*)

[55] L. Hetel, E. Fridman, and T. Floquet, *Variable structure control with generalized relays: A simple convex optimization approach*, Automatic Control, IEEE Transactions on, 60 (2015), pp. 497–502. (*Cited on p. 151.*)

[56] A. C. Hindmarsh, *ODEPACK, a systematized collection of ODE solvers*, in Scientific Computing, R. S. Stepleman et al., eds., vol. 1, North-Holland, Amsterdam, 1983, pp. 55–64. (*Cited on p. 102.*)

[57] D. Hinrichsen and A. J. Pritchard, *Mathematical Systems Theory I*, vol. 48 of Texts in Applied Mathematics, Springer Berlin Heidelberg, 2005. (*Cited on p. 175.*)

[58] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Grundlehren Text Editions, Springer Berlin Heidelberg, 2001. (*Cited on pp. 163, 164, and 165.*)

[59] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991. (*Cited on p. 175.*)

[60] D. Hua and P. Lancaster, *Linear matrix equations from an inverse problem of vibration theory*, Linear Algebra and its Applications, 246 (1996), pp. 31–47. (*Cited on p. 29.*)

[61] S. Hui and S. Żak, *On discrete-time variable structure sliding mode control*, Systems & Control Letters, 38 (1999), pp. 283–288. (*Cited on p. 16.*)

[62] J. D. Hunter, *Matplotlib: a 2D graphics environment*, Computing in Science & Engineering, (2007), pp. 90–95. (*Cited on pp. 82 and 109.*)

[63] U. Itkis, *Control Systems of Variable Structure*, Wiley New York, 1976. (*Cited on p. 1.*)

[64] M. Jean and J. J. Moreau, *Dynamics in the presence of unilateral contacts and dry friction: A numerical approach*, in Unilateral Problems in Structural Analysis — 2, G. Del Piero and F. Maceri, eds., International Centre for Mechanical Sciences, Springer Vienna, 1987, pp. 151–196. (*Cited on p. 2.*)

[65] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python*. (*Cited on p. 144.*)

[66] A. Kastner-Maresch, *Implicit Runge-Kutta methods for differential inclusions*, Numerical Functional Analysis and Optimization, 11 (1990), pp. 937–958. (*Cited on p. 16.*)

[67] A. Kastner-Maresch, *The implicit midpoint rule applied to discontinuous differential equations*, Computing, 49 (1992), pp. 45–62. (*Cited on p. 16.*)

[68] C. G. Khatri and S. K. Mitra, *Hermitian and nonnegative definite solutions of linear matrix equations*, SIAM Journal on Applied Mathematics, 31 (1976), pp. 579–585. (*Cited on p. 29.*)

[69] A. Koshkouei and A. Zinober, *Sliding mode control of discrete-time systems*, Journal of Dynamic Systems, Measurement, and Control, 122 (2000), p. 793. (*Cited on p. 16.*)

[70] U. Kotta, S. Sarpturk, and Y. Istefanopulos, *Comments on "On the stability od discrete-time sliding mode control systems"*, Automatic Control, IEEE Transactions on, 34 (1989), pp. 1021–1022. (*Cited on p. 15.*)

[71] N. Lai, C. Edwards, and S. Spurgeon, *On output tracking using dynamic output feedback discrete-time sliding-mode controllers*, Automatic Control, IEEE Transactions on, 52 (2007), pp. 1975–1981. (*Cited on p. 16.*)

[72] F. Lempio, *Euler's method revisited*, in Proc. Steklov Inst. Math., vol. 211, 1995, pp. 473–494. (*Cited on p. 16.*)

[73] A. Levant, *Sliding order and sliding accuracy in sliding mode control*, International Journal of Control, 58 (1993), pp. 1247–1263. (*Cited on pp. 1 and 8.*)

[74] A. Levant, *Robust exact differentiation via sliding mode technique*, Automatica, 34 (1998), pp. 379–384. (*Cited on pp. 134 and 160.*)

[75] A. Levant, *Higher-order sliding modes, differentiation and output-feedback control*, International Journal of Control, 76 (2003), pp. 924–941. (*Cited on p. 8.*)

[76] A. Levant, *Chattering analysis*, Automatic Control, IEEE Transactions on, 55 (2010), pp. 1380–1389. (*Cited on p. 12.*)

[77] Q. Li, *Large-Scale Computing for Complementarity and Variational Inequalities*, PhD thesis, University of Wisconsin–Madison, 2010. (*Cited on p. 94.*)

[78] D. Liberzon and S. Trenn, *The bang-bang funnel controller for uncertain nonlinear systems with arbitrary relative degree*, Automatic control, IEEE Transactions on, 58 (2013), pp. 3126–3141. (*Cited on p. 4.*)

[79] C.-F. Lin and W.-C. Su, *A total chattering-free sliding mode control for sampled-data systems*, in American Control Conference, Proceedings of the 2004, IEEE, 2004, pp. 1940–1945. (*Cited on pp. 16 and 34.*)

[80] M. Ljesnjanin, B. Draženović, Č. Milosavljević, and B. Veselic, *Disturbance compensation in digital sliding mode*, in EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE, IEEE, 2011, pp. 1–4. (*Cited on p. 77.*)

[81] Y. Lootsma, A. Van der Schaft, and K. Çamlibel, *Uniqueness of solutions of linear relay systems*, Automatica, 35 (1999), pp. 467–478. (*Cited on pp. 16 and 26.*)

[82] C. Lubich, E. Hairer, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, second ed., 2006. (*Cited on p. 14.*)

[83] Z.-Q. Luo and P. Tseng, *A decomposition property for a class of square matrices*, Applied Mathematics Letters, 4 (1991), pp. 67–69. (*Cited on p. 25.*)

[84] Č. Milosavljević, *General conditions for the existence of a quasi-sliding mode on the switching hyperplane in discrete variable structure systems*, Automation and Remote Control, 46 (1985), pp. 307–314. (*Cited on pp. 16 and 38.*)

[85] D. Mitic and Č. Milosavljević, *Sliding mode-based minimum variance and generalized minimum variance controls with $O(T^2)$ and $O(T^3)$ accuracy*, Electrical Engineering (Archiv fur Elektrotechnik), 86 (2004), pp. 229–237. (*Cited on p. 16.*)

[86] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49. (*Cited on p. 107.*)

[87] J. J. Moreau, *Unilateral contact and dry friction in finite freedom dynamics*, in Nonsmooth Mechanics and Applications, vol. 302 of International Centre for Mechanical Sciences, Springer Vienna, 1988, pp. 1–82. (*Cited on p. 102.*)

[88] Y. Orlov, *Finite time stability and robust control synthesis of uncertain switched systems*, SIAM Journal on Control and Optimization, 43 (2005), pp. 1253–1271. (*Cited on pp. 19 and 68.*)

[89] F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley Series in Nonlinear Science, John Wiley & Sons, 1996. (*Cited on p. 97.*)

[90] R. R. Phelps, *Lectures on Maximal Monotone Operators*, arXiv:math/9302209 [math.FA], (1993). (*Cited on p. 167.*)

[91] A. Pogromsky, W. Heemels, and H. Nijmeijer, *On solution concepts and well-posedness of linear relay systems*, Automatica, 39 (2003), pp. 2139–2147. (*Cited on p. 16.*)

[92] A. Polyakov and A. Poznyak, *Invariant ellipsoid method for minimization of unmatched disturbances effects in sliding mode control*, Automatica, 47 (2011), pp. 1450–1454. (*Cited on p. 7.*)

[93] S. Qu, X. Xia, and J. Zhang, *Dynamical behaviors of an Euler discretized sliding mode control systems*, Automatic Control, IEEE Transactions on, 59 (2014), pp. 2525–2529. (*Cited on p. 40.*)

[94] R. T. Rockafellar, *Convex Analysis*, vol. 28, Princeton University Press, 1997. (*Cited on pp. 163, 165, and 166.*)

[95] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317 of Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, 2009. (*Cited on pp. 28, 163, 167, and 168.*)

[96] J. Rulla, *Error analysis for implicit approximations to solutions to Cauchy problems*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 68–87. (*Cited on pp. 17, 28, and 39.*)

[97] S. Sarpturk, Y. Istefanopulos, and O. Kaynak, *On the stability of discrete-time sliding mode control systems*, Automatic Control, IEEE Transactions on, 32 (1987), pp. 930–932. (*Cited on pp. 15 and 38.*)

[98] S. Sesmat and S. Scarvarda, *Static characteristics of a three way electropneumatic servovalve*, in Proceedings of the 12th Conference on Fluid Power Technology, Aachen, Germany, 1996, pp. 643–645. (*Cited on p. 130.*)

[99] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*, Control Engineering, Birkhäuser, 2014. (*Cited on pp. 1, 8, and 12.*)

[100] Y. Shtessel, M. Taleb, and F. Plestan, *A novel adaptive-gain* supertwisting *sliding mode controller: Methodology and application*, Automatica, 48 (2012), pp. 759–769. (*Cited on p. 129.*)

[101] H. Sira-Ramirez, *Non-linear discrete variable structure systems in quasi-sliding mode*, International Journal of control, 54 (1991), pp. 1171–1187. (*Cited on p. 16.*)

[102] E. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6 of Textbooks in Applied Mathematics, Springer New York, second ed., 1998. (*Cited on p. 73.*)

[103] W.-C. Su, S. Drakunov, and Ü. Özgüner, *Implementation of variable structure control for sampled-data systems*, in Robust Control via Variable Structure and Lyapunov Techniques, Springer, 1996, pp. 87–106. (*Cited on p. 16.*)

[104] W.-C. Su, S. Drakunov, and Ü. Özgüner, *An $O(T^2)$ boundary layer in sliding mode for sampled-data systems*, Automatic Control, IEEE Transactions on, 45 (2000), pp. 482–485. (*Cited on pp. 15 and 77.*)

[105] Y. Z. Tsypkin, *Theory of Relay Control Systems*, Moscow: Gostechizdat, 1955. (*Cited on p. 1.*)

[106] V. Utkin, *Variable structure systems with sliding modes*, Automatic Control, IEEE Transactions on, 22 (1977), pp. 212–222. (*Cited on p. 1.*)

[107] V. Utkin, *Sliding Modes in Control and Optimization*, Communications and Control Engineering, Springer Berlin, 1992. (*Cited on pp. 1, 7, 35, 38, and 70.*)

[108] V. Utkin, *Sliding mode control in discrete-time and difference systems*, in Variable Structure and Lyapunov Control, vol. 193 of Lecture Notes in Control and Information Sciences, Springer, 1994, pp. 87–107. (*Cited on pp. 15 and 34.*)

[109] V. Utkin and H. Lee, *The chattering analysis*, in Power Electronics and Motion Control Conference, EPE-PEMC 12th International, IEEE, 2006, pp. 2014–2019. (*Cited on p. 12.*)

[110] B. Walcott and S. H. Żak, *Observation of dynamical systems in the presence of bounded nonlinearities/uncertainties*, in Decision and Control, 1986 25th IEEE Conference on, vol. 25, IEEE, December 1986, pp. 961–966. (*Cited on p. 75.*)

[111] B. Wang, *On discretization of sliding mode control systems*, PhD thesis, RMIT University, 2008. (*Cited on p. 16.*)

[112] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, *Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: Towards chattering suppression in output and input signals*, in Variable Structure Systems (VSS), 2014 13th International Workshop on, June 2014, pp. 1–6. (*Cited on pp. 130 and 138.*)

[113] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, *Comparisons between implicit and explicit discrete-time implementations of equivalent-control-based sliding mode controllers: input and output chattering suppression via the implicit method*, Control Systems Technology, IEEE Transactions on, (2015), p. in press. (*Cited on p. 147.*)

[114] B. Wang, X. Yu, and X. Li, *ZOH discretization effect on higher-order sliding-mode control systems*, Industrial Electronics, IEEE Transactions on, 55 (2008), pp. 4055–4064. (*Cited on p. 110.*)

[115] L. Westphal, *Lessons from an example in "On the stability of discrete-time sliding mode control systems"*, Automatic Control, IEEE Transactions on, 44 (1999), pp. 1444–1445. (*Cited on p. 15.*)

[116] X. Yu, *Sliding-mode control with soft computing: A survey*, Industrial Electronics, IEEE Transactions on, 56 (2009), pp. 3275–3285. (*Cited on p. 16.*)

[117] X. Yu, B. Wang, Z. Galias, and G. Chen, *Discretization effect on equivalent control-based multi-input sliding-mode control systems*, Automatic Control, IEEE Transactions on, 53 (2008), pp. 1563–1569. (*Cited on pp. 12 and 16.*)